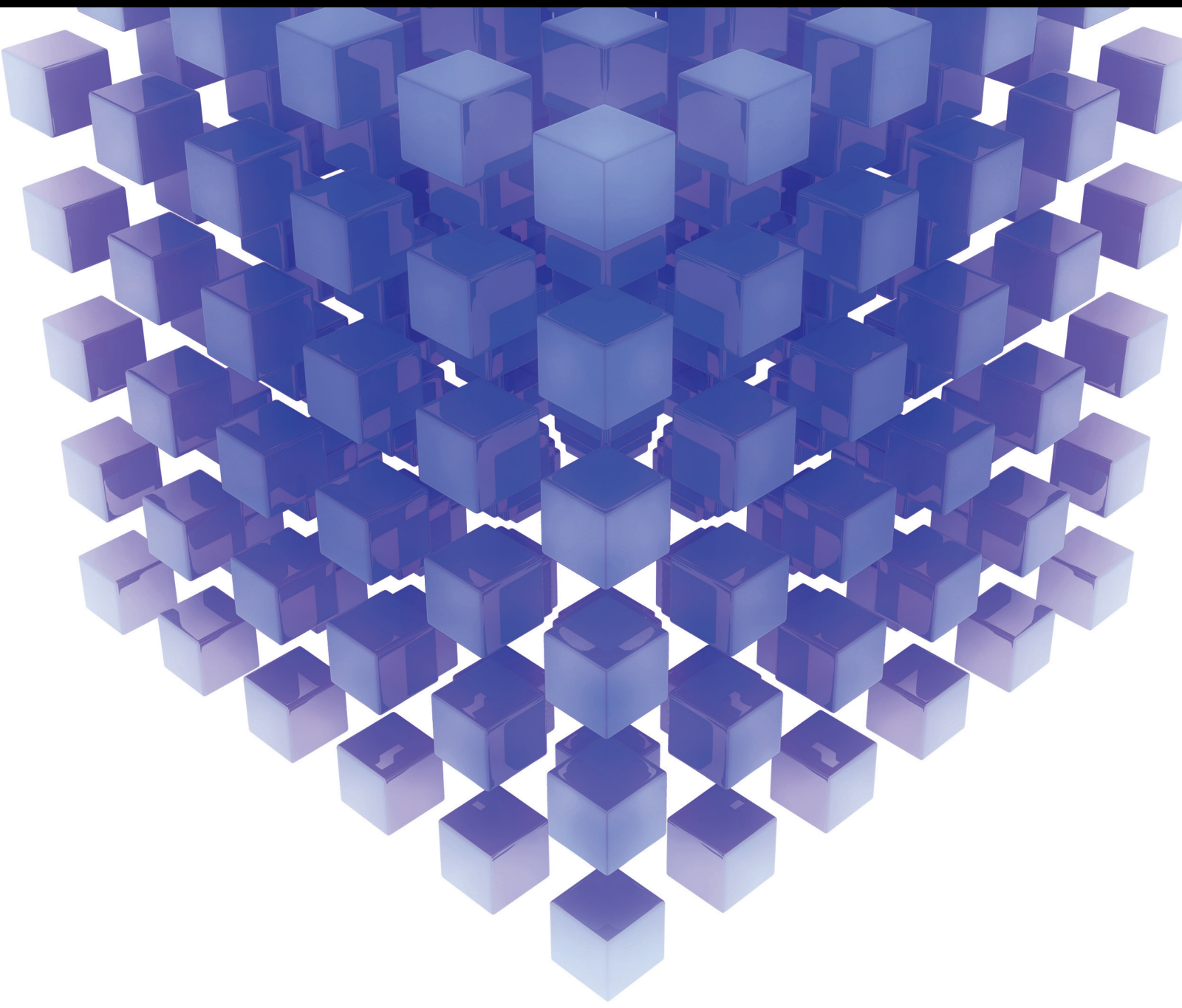
Mathematical Problems in Engineering

Advanced Cloud Computing and Novel Applications

Guest Editors: Bao Rong Chang, Ngoc Thanh Nguyen, Bay Vo, and Hui-Huang Hsu



Advanced Cloud Computing and Novel Applications

Mathematical Problems in Engineering

Advanced Cloud Computing and Novel Applications

Guest Editors: Bao Rong Chang, Ngoc Thanh Nguyen, Bay Vo, and Hui-Huang Hsu

Editorial Board

Contents

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 630923, 2 pages
http://dx.doi.org/10.1155/2015/630923

Editorial
Advanced Cloud Computing and Novel Applications

Bao Rong Chang,[1] Ngoc Thanh Nguyen,[2] Bay Vo,[3] and Hui-Huang Hsu[4]

[1]*Department of Computer Science & Information Engineering, National University of Kaohsiung, Kaohsiung 81148, Taiwan*
[2]*Institute of Informatics, Wroclaw University of Technology, 50370 Wroclaw, Poland*
[3]*Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City 70000, Vietnam*
[4]*Department of Computer Science & Information Engineering, Tamkang University, New Taipei City 25137, Taiwan*

Correspondence should be addressed to Bao Rong Chang; brchang@nuk.edu.tw

Received 13 July 2015; Accepted 22 July 2015

The study on advanced cloud computing encompasses a large number of the most important as well as promising direction for scientific research and development in the era of advanced novel applications which explores the fundamental roles and interactions as well as practical impacts of advanced computing and information technologies on the next generation of cloud-empowered products, systems, services, and activities. Advanced information technology and computing engineering have involved internet of things, ubiquitous computing, social networks, and data/knowledge grids, as well as cloud computing and service-oriented architecture.

A lot of relevant topics have been raised in this issue, though many other topics could be concerned. This special issue is intended to foster a high profile, leading edge forum for people, either in the scope of R&D projects, engineering, or business applications, who are working in the field to contribute and disseminate innovative new work on advanced cloud computing. Here high quality technical papers in all aspects of advanced cloud computing including emerging trends and applications, theoretical studies, and experimental prototypes have been presented.

Z. Peng et al. deal with cloud computing resources provisioning based on reinforcement learning. With the introduction of the concepts of segmentation service level agreement (SSLA) and utilization unit time cost (UUTC), this study designed a novel optimization object function and employed reinforcement learning to solve resources provisioning. L. Pan and D. Wang develop metaheuristic solutions based on the genetic algorithm to tickle IaaS cloud provider revenue maximization (ICPRM) problem. The paper proposed by S. Li et al. gives a novel multilevel data encoding scheme satisfying these requirements in mobile cloud computing applications, particularly in the field of telerehabilitation.

The security management is addressed in three aspects. K. Mazur et al. implemented the role-based access control method in quality of protection modeling language (QoP-ML) and evaluated its performance in terms of mentioned factors to determine the most secure, energy-efficient, environmental-friendly security mechanisms. In cloud computing people really care user data protection through identity management; L. Wu et al. introduce a reputation mechanism and design a reputation-based identity management model for cloud computing to solve these problems. In order to guarantee the credibility of this information, D. Wang et al. present a novel trust degree model of the credibility information and then propose a new composite semantic web service selection approach based on this credible information.

Besides those, there are several interesting topics in the issue. C.-J. Tang and M.-R. Dai suggest modeling and analysis of queueing-based vary-on/vary-off (VOVO) schemes for server clusters. X. Liu et al. preform wireless-uplinks-based energy-efficient scheduling in mobile cloud computing. C.-L. Chou et al. realize multicriteria resource brokering in cloud computing for streaming service. Y. Liu and W. Wei establish a replication-based mechanism for fault tolerance in MapReduce framework.

By compiling these papers, we hope to enrich our readers and researchers with respect to the various problems and their solutions in cloud computing and novel applications.

Bao Rong Chang
Ngoc Thanh Nguyen
Bay Vo
Hui-Huang Hsu

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 238245, 15 pages
http://dx.doi.org/10.1155/2015/238245

Research Article

A Reputation-Based Identity Management Model for Cloud Computing

Lifa Wu,[1] Shengli Zhou,[1,2] Zhenji Zhou,[1] Zheng Hong,[1] and Kangyu Huang[1]

[1]*College of Command Information System, PLAUST, Nanjing 210007, China*
[2]*Information Department, Zhejiang Police College, Hangzhou 310000, China*

Correspondence should be addressed to Shengli Zhou; 76933768@qq.com

Received 21 March 2015; Revised 27 May 2015; Accepted 30 May 2015

Academic Editor: Bao Rong Chang

In the field of cloud computing, most research on identity management has concentrated on protecting user data. However, users typically leave a trail when they access cloud services, and the resulting user traceability can potentially lead to the leakage of sensitive user information. Meanwhile, malicious users can do harm to cloud providers through the use of pseudonyms. To solve these problems, we introduce a reputation mechanism and design a reputation-based identity management model for cloud computing. In the model, pseudonyms are generated based on a reputation signature so as to guarantee the untraceability of pseudonyms, and a mechanism that calculates user reputation is proposed, which helps cloud service providers to identify malicious users. Analysis verifies that the model can ensure that users access cloud services anonymously and that cloud providers assess the credibility of users effectively without violating user privacy.

1. Introduction

Cloud computing is a computing model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and other services) that can be rapidly provisioned and released with minimal management effort or cloud service provider (CSP) interaction [1]. Large numbers of users may simultaneously engage in cloud computing services, making the "multitenant" feature an important property of cloud computing according to the Cloud Security Alliance (CSA) [2]. However, the multitenant property brings the following new problems:

(i) Privacy leaks because of external user data: in an open environment, users must be authenticated to access cloud services. If users employ their actual names (or fixed usernames) to log in, sensitive information such as login names or even long-term behavior may be revealed by data mining or other techniques and used illegally by the CSP.

(ii) Management problems caused by an excessive number of tenants: to preserve privacy, users must employ different pseudonyms to access each cloud computing session. In this case, it is difficult for users to recall a large number of pseudonyms and passwords. Meanwhile single-use pseudonyms make no contribution to the development of a user's reputation.

(iii) Security threats to the CSP caused by multiple tenants: by accessing cloud services, malicious users may take the opportunity to attack the CSP through activities such as stealing data, performing vulnerability scanning, and launching denial of service (DoS) attacks. In particular, if allowed to log in by a pseudonym, malicious users can launch whitewash attacks, where the malicious user can continue to visit the CSP normally after an attack or initiate another attack by creating a new pseudonym.

Considering the above problems, traditional identity management mechanisms that store user identities in a database directly are no longer applicable [3]. In this paper, we design a new identity management model based on user reputation, which is herein denoted as reputation-based identity management (RIM). The main contributions of RIM are listed as follows:

(i) Anonymous user access: in RIM, we design a method in which each user takes a different pseudonym for each session when accessing cloud services. No link between a user identity and a corresponding pseudonym is provided, and no link is provided between the pseudonyms of a single user. Pseudonym usage does not affect user attestation, and it decreases the input of private user information, rendering it impossible for tenants to spy on each other.

(ii) User attestation: in our model, users firstly register with an identity provider (IdP), which provides the user with a formal identity certificate. The identity certificate is the basis for the user to prove their legitimate status to the CSP.

(iii) Reputation attestation: RIM records and determines user reputations as user activities accumulate with access to successive cloud computing sessions and provides proof of the reputation. The proof assures the credibility of the reputation, ensuring that the reputation indeed belongs to its owner. The proof also guarantees nonrepudiation; that is, a user cannot deny the reputation assigned to them. As such, the proof ensures unforgeability. Users cannot promote their reputation without the authorization of the IdP. The introduction of a reputation does not affect the anonymity of users.

The remainder of this paper is organized as follows. Section 2 describes related work. Section 3 introduces the background and technology of cloud computing along with identity management. Section 4 introduces the proposed RIM and describes its design and realization. Section 5 analyzes the correctness and security of our model. Finally, the last section concludes the paper and proposes future work.

2. Related Work

The focus of identity management for cloud computing is user privacy. In [4], the authors proposed an approach to preserve the privacy of users based on zero-knowledge proof protocols and semantic matching techniques. The approach also enhanced the interoperability across multiple domains. Although the study realized the goal of concealing user identities, the CSP could still obtain sensitive information through data mining techniques because user identities were consistent throughout the entire process. In [5], the authors proposed an entity-centric approach for identity management in cloud computing. The approach was based on personally identifiable information (PII) and on anonymous identification to mediate interactions between users and cloud services. Although the identification was anonymous, the PII released privacy related information. In [6], the authors took advantage of attribute-based encryption and signature technology to conceal user identities in cloud computing. However, because users must employ a single unchanging certificate to obtain authentication from the CSP, an attacker may ascertain a user's identity easily through the static certificate. In [7], the authors improved the approach proposed in [4], where the registry center was not required to be online at all times. However, the improved approach had the same disadvantages with respect to user privacy protection as the original approach. In [8, 9], the authors designed and introduced a method that integrated blind signature and hash chain techniques to protect user privacy in cloud computing. However, in every session, users employ the same pseudonym to log in, which results in linkability between different sessions. Although the above studies achieve the goal of user identity concealment, the successive behaviors of individual users can be associated. Thus, through analyzing user behaviors, the CSP can potentially compromise user privacy. In [10], the authors proposed a method for anonymity using group digital signature technology. Because this method introduced no correlation between user signatures, this method is an improvement over the aforementioned methods. However, the use of group digital signature technology forbids members from joining and leaving a group dynamically, which conflicts with the openness of cloud computing.

In this paper, we propose an approach that achieves user identity anonymity. Moreover, by this approach, the CSP is unable to link user behavior to user identity. While the anonymity of identity protects user privacy, it tends to enable CSP attack by malicious users because the CSP is unable to trace the users involved in the attack. Therefore, our approach introduces trust management to address the shortcomings of identity anonymity. The approach determines user reputations and binds the reputation to user identity. The CSP utilizes user reputation to distinguish malicious users and to reduce the threat of attack by malicious users [11].

Previous reputation research has focused on peer to peer (P2P) systems, which has been developed into various systems, for example, the EigenTrust [12], the PeerTrust [13], and the PowerTrust [14] systems. Along with the development of electronic business, reputation research has had a service oriented focus, where the reputation of service providers has been evaluated to protect consumers. In [15, 16], reputation was employed as a means of choosing service providers. In cloud computing, most research has focused on protecting the CSP, such as what was done in [17, 18]. However, the use of a reputation model to manage users in cloud computing, such as that employed in this paper, has not been the subject of previous research. In [19], the authors designed an identity management model for a noncloud computing environment. The model supposed that users must be previously registered, which conflicts with the required openness of cloud computing. In [20, 21], the authors designed a reputation-based identity management model that used online systems for applications in, for example, electronic business and forums. The online system must recognize user identities or unique identifiers to accumulate reputation data. However, tracking user identities tends to leak user information, which threatens privacy. This is why these approaches cannot be applied to cloud computing. Our proposed RIM introduces reputation to manage users and simultaneously protects user privacy, which has not been investigated by other researchers.

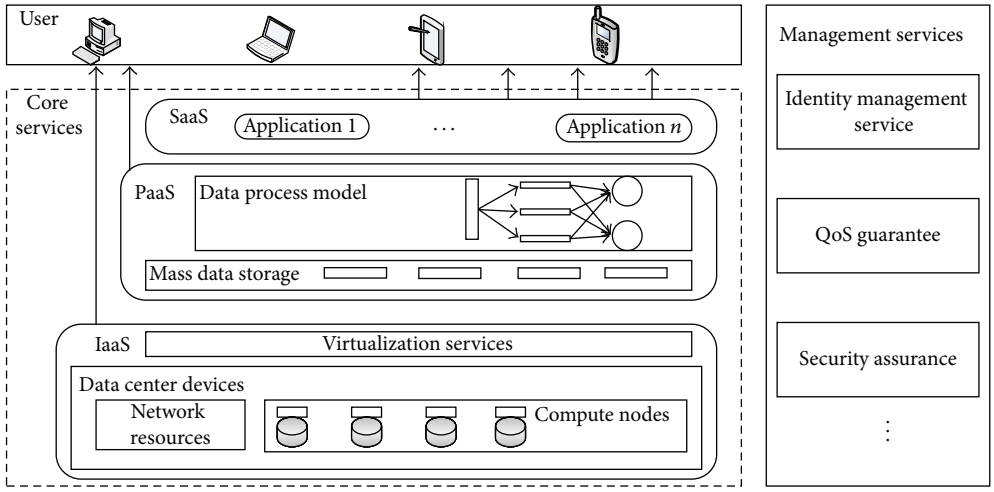


Figure 1: Cloud services architecture.

User privacy protection requires a user identity to be anonymous such that it cannot be tracked. However, the determination of reputation must confirm user identity and track user transaction history. Therefore, reputation management and identity anonymity appear to be contradictory [22]. In [22], the authors alleviated this contradiction and designed an anonymous reputation system in a P2P environment using electronic cash technology to provide feedback to users according to behavior. However, according to this approach, the amount of feedback one user gives to another is restricted by the quantity of electronic coins in the former user's possession. A user with no electronic coins is therefore barred from involvement in the transaction. As such, this method contradicts with the openness of cloud computing. In [23], the authors used blind signature technology to achieve a reputation-based identity management approach in the C-S mode. This approach provides the service provider with user reputations while guaranteeing user anonymity. However, under this approach, a service provider can confirm that orders derive from the same user and accumulate the history of user actions, resulting in the potential violation of user privacy. In [24], authors achieved an anonymous reputation system based on zero-knowledge authentication and digital signature technology. However, [4] the sessions between the user and service provider under this approach are linkable, enabling service providers to violate user privacy. From the above analysis, we can see that these last two examples [25] manage to provide user reputation only by divulging user identity. Our RIM overcomes these shortcomings by determining user reputation while ensuring that user identity and pseudonyms are unlinkable.

3. Technology Background

3.1. Cloud Computing Architecture. Cloud computing provides three main service delivery models to the public, including the following [2]:

(i) Software as a Service (SaaS): this is a software delivery model in which software and its associated data are hosted in the cloud and are typically accessed by users using a thin client.

(ii) Platform as a Service (PaaS): this is the delivery of a computing platform and solution stack as a service, which provides all the facilities required to support the complete life cycle of building and delivering web applications and services from the Internet.

(iii) Infrastructure as a Service (IaaS): this delivers computer infrastructure (typically a platform virtualization environment) as a service, along with raw storage and networks. Rather than purchasing servers, software, data-center space, or network equipment, clients purchase the resources as a fully outsourced service.

Based on the above approach, a number of similar abstract service models have been recently promoted, such as Hardware as a Service (HaaS), Data as a Service (DaaS), and Communication as a Service (CaaS). Cloud computing management services ensure the reliability, availability, and security of core services. Figure 1 [26, 27] illustrates the basic service architecture of cloud computing.

Cloud identity management takes charge of identity management throughout the entire cloud services stack. Identity management can be divided into three categories [28]: isolated user identity model, federated user identity model, and centralized user identity model, which are defined as follows:

(i) Isolated user identity model: in this model, the service provider acts as both credential provider and identifier provider to their clients. A user obtains separate unique identifiers from each service/identifier provider transacted with.

(ii) Federated user identity model: this can be defined as the set of agreements, standards, and technologies that enable a group of service providers to recognize user identifiers and entitlements from other service providers within a federated domain.

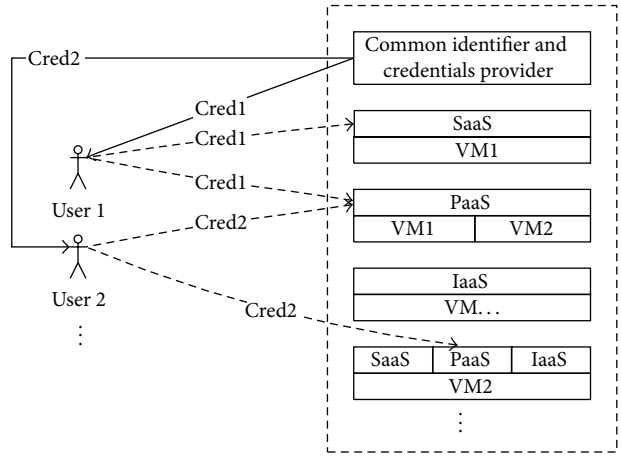


Figure 2: Centralized user identity model in cloud computing.

(iii) Centralized user identity model: this model employs a single identifier and credential provider that is used by all service providers to provide identifiers and credentials to users.

In the centralized user identity model, the job of identity management has been taken over by an identity provider from the service provider. This reduces the burden of the service provider and also reduces the number of certificates required by users to hold. In addition, the centralized user identity model facilitates the delivery of the identity management service from the CSP to the users. In an identity provider management domain, users first obtain a formal certificate and then access cloud services in the same domain using this certificate. This property makes the centralized user identity model suitable for cloud computing, and, therefore, we employ a centralized model in our design, as shown in Figure 2.

3.2. Basic Technology. We introduce identity-based signature, blind signature, and zero-knowledge authentication technology to implement RIM.

The identity-based signature was proposed by Shamir [29] in 1984. Here, a user's identity is first disclosed as a public key and then used to generate a private key. In [29], the authors designed a conceptual model but did not provide for any real implementation. Since then, studies of the identity-based signature have been conducted, but no efficient and provable signature scheme was concluded until Boneh et al. [30, 31] promoted an identity-based encryption scheme. Identity-based signature comprises four stages: setup, private key extraction, signature, and verification. Using an identity as a public key has a natural legitimacy, which can simplify the process of key distribution.

The concept of blind signature was suggested by Chaum [32]. The approach ensures that the signer does not know the specific content of the message and that the message owner can obtain the signature of the message. Blind signature consists of three operations: blinding, signing, and blindness removal. Blind signature has many characteristics conducive to protecting user privacy such as (1) blindness, where the signer does not know the specific content of the message to be signed and (2) untraceability, where, if a signature is leaked, the signer has no idea of when and by whom the message was signed.

Zero-knowledge proof authentication technology refers to a prover's assurance of ownership of some secret information, without revealing any useful knowledge about the secret information to the verifier. Well-known zero-knowledge authentication schemes mainly include the Feige-Fiat-Shamir [33] scheme, the Guillou-Quisquater [34] scheme, and the Schnorr [35] scheme.

The proposed RIM achieves an identity management model that is suitable for cloud computing by employing methods related to the above techniques that are most suitable for specific application scenarios. In Section 4.3, we give a detailed exposition of our design.

4. Model Design and Implementation

In this section we firstly give some assumptions for RIM and then explain the detailed implementation of the model.

4.1. Assumptions. We must make some reasonable assumptions to ensure proper RIM functionality. Firstly, we assume the existence of an independent arbitration agency whose function is similar to a governmental authority that can ensure user privacy while simultaneously providing a secure source of anonymous user information under very specific circumstances. As such, the agency proposed herein is different from a trusted third party in an ordinary sense. The agency need not have a continual online presence. Moreover, the agency does not collude with other users, and user information is protected from malicious users. Because cloud services are open to the public as a paid service, it is necessary to reveal anonymous user information when a CSP suffers an attack or has economic disputes with users. Therefore, RIM must be able to identify anonymous users to address these specific problems. If a CSP seeks anonymous information, it must provide a range of certificates and follow prescribed security protocols to ensure that the application submitted is legitimate.

Another important assumption is that the communication channel in RIM is secure. SSL and IPSec protocols can be used to ensure the security of the channel.

4.2. Model Design. Four roles are managed in RIM: User, CSP, IdP, and deanonymizing authority (DA), which are defined as follows:

(i) User: the consumer of cloud services who requests identity anonymity and benefits from the service.

(ii) CSP: the service provider who, after the transaction between the User and CSP, provides feedback regarding the transaction.

(iii) IdP: the service provider who not only provides registration services to the User but also determines the User reputation, which is indicative of the degree

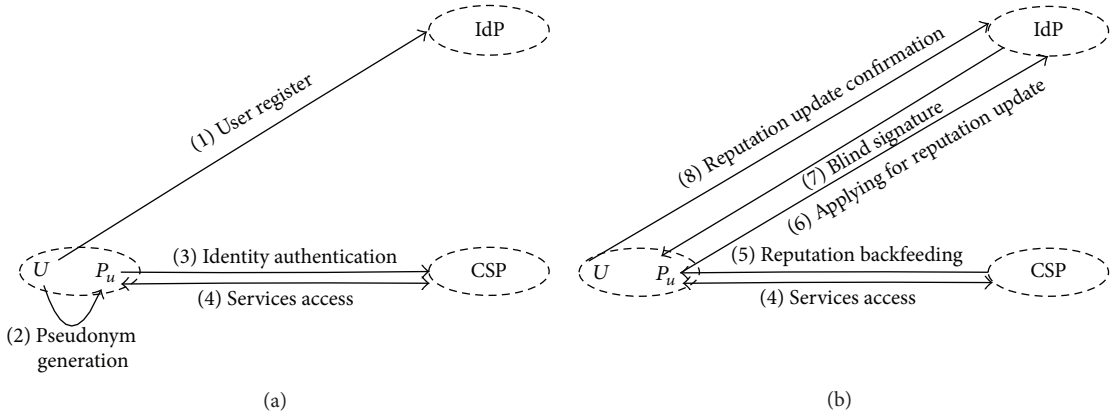


Figure 3: Schematic of the anonymous access and reputation update process.

of trust, based on the feedback obtained from the CSP. The IdP issues the User reputation certificate.

(iv) DA: an authority that can reveal User pseudonyms and provide User identity-related information to the CSP.

RIM has five stages, namely, Environment Initialization, User Registration, Identity Authentication, Reputation Computation, and Pseudonym Disclosure.

In the Environment Initialization stage, RIM first creates the public parameters of the IdP, DA, and CSP. It then generates the public and private keys used for signatures. The operations of key generation are as follows: λ is the security parameter, PK denotes the public key, and SK denotes the private key. The following describes the probabilistic polynomial time algorithm:

(i) $(PK_{Id\_s}, SK_{Id\_s}) \leftarrow Setup\_IdP\_Idsign(1^{\lambda})$: it is a key-generation algorithm that takes λ as the input and outputs a pair (PK_{Id_s}, SK_{Id_s}) of public and secret keys used by the IdP to sign the identity certificate.

(ii) $(PK_{Id\_b}, SK_{Id\_b}) \leftarrow Setup\_IdP\_blind(1^{\lambda})$: given the security parameter λ, it creates a pair of keys (PK_{Id_b}, SK_{Id_b}) which is used by the IdP to generate a blind signature.

(iii) $(PK_{CSP}, SK_{CSP}) \leftarrow Setup\_CSP(1^{\lambda})$: it outputs a pair of keys (PK_{CSP}, SK_{CSP}) which is used by the CSP to create a feedback certificate.

(iv) $(PK_{DA}, SK_{DA}) \leftarrow Setup\_DA(1^{\lambda})$: it gives the DA a pair of keys (PK_{DA}, SK_{DA}) which is used to encrypt and decrypt User identities.

Keys (PK_{Id_s}, SK_{Id_s}) and (PK_{Id_b}, SK_{Id_b}) can be combined into a single pair. However, in such a situation, if the keys are disclosed, the IdP will collapse. When they are separated, if one pair of keys is compromised, for example, if the blind signature key is compromised, only the credibility of the reputation is affected while the identity certificate remains valid. Owing to security considerations, we maintain these two key pairs as separate.

The Registration operation and Verification operation contained in the User Registration stage are defined as follows:

(i) Registration: the User initiates the operation $(C_{Id}, rep, C_{rep}) \leftarrow Register(SK_{Id\_s}, Id)$ to register with the IdP. The User obtains an identity (Id) as the input and then receives the identity certificate C_{rep}, reputation (rep), and reputation certificate C_{rep}. Here, rep is the initial reputation value for new users given by the IdP.

(ii) Verification: the operation $1/\perp \leftarrow CheckReg(PK_{Id\_s}, C_{Id}, rep, C_{rep})$ is employed by the User to authenticate C_{Id}, rep, and C_{rep}. This operation takes the IdP public key PK_{Id_s} as one of the inputs, which will confirm the legitimate source of these certificates. If verified, 1 will be returned as the result; otherwise, $\perp$ will be returned.

The first time a User enters the cloud service system, the User must register with the IdP using their own identity. The IdP will determine whether the identity is redundant or on the blacklist. If the answer is no, the verification is passed. The IdP gives the User an initial reputation value and issues an identity certificate and reputation certificate. The identity that registered to the IdP is known to the public. It can be a URL or e-mail address associated with the User. The public identity carries less privacy and is easily verified by the IdP. Because of the openness of cloud services, the rules for accessing cloud services cannot be overly strict. In RIM, all users that meet the basic safety requirements are allowed access. After Registration, the Verification operation is performed to verify the identity certificate and the reputation certificate. The above processes are illustrated in Step 1 of Figure 3(a).

After the User has a recognizable valid identity, access is granted to the CSP using the pseudonym generated according to this valid identity, and the CSP will authenticate the pseudonym. The Identity Authentication stage includes the Pseudonym Generation operation and Authentication operation shown in steps 2 and 3 of Figure 3(a), which are described as follows:

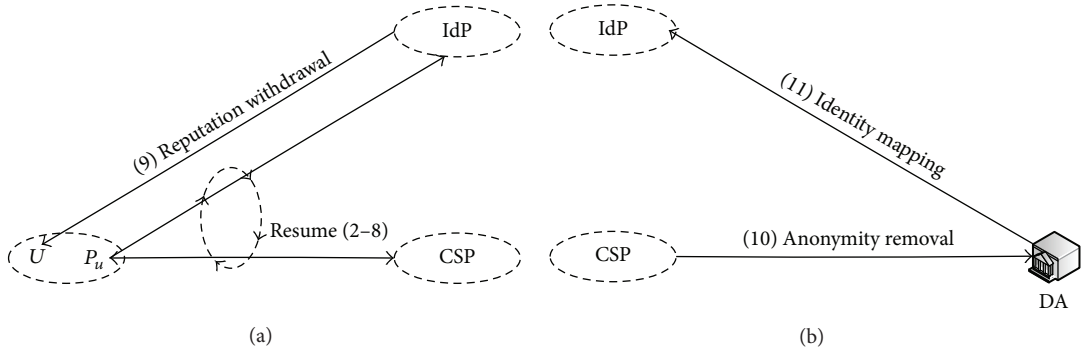


Figure 4: Schematic of the standard access process and Pseudonym Disclosure.

(i) Pseudonym Generation: taking Id, the User identity certificate C_{Id}, and rep as the inputs, the operation $P_u \leftarrow Gen\_pny(Id, C_{Id}, rep)$ outputs the User pseudonym P_u, which is the only identification presented to the CSP.

(ii) Authentication: the operation $1/\perp \leftarrow Authenticate(P_u, C_{rep})$ is used to authenticate the User identity through P_u by the CSP. This operation is also used to confirm the credibility of the User reputation.

The Pseudonym Generation operation encrypts Id using the public key provided by the DA. As such, if a dispute arises between the User and the CSP, RIM would decrypt P_u with the private key provided by the DA to restore the User Id. The Authentication operation applies the Σ-protocol to conduct the authentication. During the process, the CSP can extract P_u, C_{rep}, and rep but not the User Id.

After the transaction between the User and the CSP, RIM enters the Reputation Computation stage, where the User rep is updated on the basis of feedback provided by the CSP. This stage includes a series of operations including the Reputation Backfeeding operation, Blinding operation, Applying for Reputation Update operation, Blind signing operation, and Reputation Update Confirmation operation, which are defined as follows:

(i) Reputation Backfeeding: the CSP calls $(rep\_new, C_{rep\_new}) \leftarrow Gran(P_u)$ and provides reputation feedback rep_new and its certificate C_{rep_new} according to the performance of P_u. The C_{rep_new} guarantees that rep_new is from the CSP.

(ii) Blinding: P_u randomly selects the blind factor $Nonce$ as the input of the operation $Nonce\_blind \leftarrow Blind(Nonce)$ to obtain the blinded value Nonce_blind.

(iii) Applying for Reputation Update: given the inputs $rep\_new, C_{rep\_new}, Nonce\_blind$, P_u calls $1/\perp \leftarrow Update(rep\_new, C_{rep\_new}, Nonce\_blind)$ to apply for reputation updating. The IdP verifies the reputation feedback based on rep_new and C_{rep_new}. A successful operation returns 1; otherwise, it returns $\perp$.

(iv) Blind Signing: the IdP calls $C_{blind} \leftarrow Blind\_sign(Nonce\_blind)$ to generate the blind signature C_{blind} of the value $Nonce\_blind$.

(v) Reputation Update Confirmation: the User removes the blindness of C_{blind} and obtains the certificate C_{Nonce} of $Nonce$. Then, the User calls $1/\perp \leftarrow Confirm\_Update(C_{Nonce})$ to submit the request for confirming the update of reputation to the IdP. If the IdP successfully updates the User rep, it returns 1; otherwise, it returns $\perp$.

The CSP provides feedback to P_u using the Reputation Backfeeding operation, as illustrated by Step 5 in Figure 3(b). The Applying for Reputation Update Confirmation operation submits the reputation feedback rep_new and a blinded random value $Nonce\_blind$ to the IdP and expects that the IdP will update rep. This operation is illustrated by Step 6 in Figure 3(b). The IdP verifies the feedback and determines that it is valid. However, the IdP cannot determine for whom the feedback is designated because the IdP cannot track the feedback to the User Id. The IdP uses the Blind Signing operation to sign $Nonce\_blind$ and gives the result C_{blind} back to P_u, as illustrated by Step 7 in Figure 3(b). Because the User generates P_u directly, the User can locate their own P_u. The User obtains C_{blind} through P_u and then removes the blindness of C_{blind} to obtain C_{Nonce}. The User subsequently calls the Reputation Update Confirmation operation and submits C_{Nonce} to the IdP to confirm the updating of rep, which is illustrated by Step 8 in Figure 3(b). The IdP verifies C_{Nonce}, and, if passed, the value of rep is updated. The IdP cannot link $Nonce$ to $Nonce\_Blind$, which is why the blind signature is introduced. This method ensures that, even in the case of collusion between the IdP and CSP, the IdP cannot obtain the User Id through P_u.

The aforementioned processes address the entire process of new user access to cloud services, including User Registration, accessing services, and reputation update. The User who is already registered must first call the Reputation Withdrawal operation to acquire a value for rep from the IdP and then follow all subsequent operations, as the aforementioned. The Reputation Withdrawal operation is illustrated in Figure 4(a), which is described as follows:

(i) Reputation Withdrawal: the User acquires rep and C_{rep} by the operation $(rep, C_{rep}) \leftarrow Withdraw(Id)$ using its own Id.

The last stage is the Pseudonym Disclosure stage. If the User has done harm to the CSP, the CSP must gain access to the User Id. This stage includes the Anonymity Removal operation and the Identity Mapping operation, which are described as follows:

(i) Anonymity Removal: when the operation $g^{Id} \leftarrow De\_Anonymity(P_u, SK_{DA})$ is called, the DA opens P_u using its private key SK_{DA}. The output of this operation is not the User Id itself but g^{Id}, which is generated by the IdP, where g is one of the public parameters.

(ii) Identity Mapping: IdP calls $Id \leftarrow Map(g^{Id})$ to map g^{Id} to the User Id and then resolves the disputes offline.

The CSP submits a malicious behavior report for the User and pseudonym P_u to the DA and applies for opening a pseudonym, as illustrated in Step 10 in Figure 4(b). The DA uses the Anonymity Removal operation to open P_u and obtains g^{Id}. The DA submits g^{Id} to the IdP. Then the IdP calls the Identity Mapping operation and retrieves the User Id, as illustrated by Step 11 in Figure 4(b). Therefore, for security considerations, the DA does not recover the Id directly. This is one of the methods that ensure user privacy and prevent the misuse of a user's identity.

The above provides an overview of RIM. The following provides a concrete realization of our model.

4.3. RIM Realization. In this section we provide a description of the concrete realization of RIM based on Hidden Identity-Based Signaturesproposed by Kiayias and Zhou [36]. Hidden Identity-Based Signatures meet user requirements for anonymity in a cloud computing environment, but they do not satisfy the need for a unique user pseudonym for each session. We therefore extended Hidden Identity-Based Signatures by introducing reputation and blind signature to fulfill the requirements mentioned above. In RIM, we achieve blind signature using the Schnorr scheme [25]. The public parameters discussed previously [25, 36] are identical, so the RIM is made more comprehensive by combining the two techniques.

In the Environment Initialization stage, we generate public parameters $\langle p, g, G, G_T, e\rangle$, where G and G_T are cyclic groups of prime order p, g is a generator for G and G_T, e is a bilinear map, $e : G \times G \rightarrow G_T$, and $|G| = |G_T|$. h is selected from $G \setminus \{1\}$ randomly. H is a hash function, $H : \{0,1\}^* \rightarrow Z_p$. The operation $Setup\_IdP\_Idsign(1^\lambda)$ randomly selects $x, y \overset{r}{\leftarrow} Z_p^*$ and computes $X = g^x$, $Y = g^y$, where the public and private key pairs are given as $PK_{Id\_s} = (X, Y)$, $SK_{Id\_s} = (x, y)$. Similarly, the operation $Setup\_IdP\_blind(1^\lambda)$ generates the key pair $PK_{Id\_b} = M$, $SK_{Id\_b} = m$, where $m \overset{r}{\leftarrow} Z_p^*$ and $M = g^{-m}$. $Setup\_CSP(1^\lambda)$ generates the key pair $PK_{CSP} = (I, J)$, $SK_{CSP} = (i, j)$ for feedback signature using the same method described above. The key pair generation of $Setup\_DA(1^\lambda)$ is more complicated. It randomly selects $u, v \overset{r}{\leftarrow} G, w \overset{r}{\leftarrow} G\backslash\{1\}$ and $b, d \overset{r}{\leftarrow} Z_p^*$, resulting in $w = u^b = v^d$, which provides the key pair $PK_{DA} = (u, v, w)$, $SK_{DA} = (b, d)$. After the keys are generated, RIM publishes the public keys and keeps the private keys secret.

In the User Registration stage, the IdP provides the User with the identity certificate, reputation, and reputation certificate. We introduce the technology of short signatures [37] to issue these certificates. The IdP randomly selects $r, q \overset{r}{\leftarrow} Z_p$ and then issues identity certificate $C_{Id} \leftarrow (g^{1/(x+Id+yr)}, r)$ and reputation certificate $C_{rep} \leftarrow (g^{1/(x+rep+yq)}, q)$, noting that Id and rep denote the identifier and reputation belonging to the User. After that, the IdP gives the User a triple $\langle C_{Id}, C_{rep}, rep\rangle$.

The User verifies these certificates using the IdP public key. When the equation $e(g^{1/(x+Id+yr)}, Xg^{Id}Y^r) = e(g, g)$ holds, the User accepts the identity certificate. In the same way, the User verifies the reputation certificate. If the two certificates are both valid, the operation $CheckReg(PK_{Id\_s}, C_{Id}, rep, C_{rep})$ returns 1. If either certificate is invalid, both certificates are discarded.

The Pseudonym Generation and Authentication operations in the Identity Authentication stage are usually carried out together in practice. In this paper, we extend the method introduced in [36] to authenticate the User identity, User identity certificate, and the link between the User identity and identity certificate. Our approach requires verification that the reputation generated from the pseudonym belongs to the User we have just authenticated. The specific process is described as follows:

(i) The User employs linear encryption [37] to commit Id in (U, V, W). It satisfies the constraints $U = u^k$, $V = v^l$, $W = w^{k+l}g^{Id}$, where l and k are randomly selected: $K, l \overset{r}{\leftarrow} Z_p$.

(ii) The User commits C_{Id} in (S, R); that is, $S = g^{r_1}C_{Id}$ and $R = g^{r_2}h^{r_1}Y^r$, where r_1 and r_2 are randomly selected: $r_1, r_2 \overset{r}{\leftarrow} Z_p$.

(iii) The User authenticates their identity using the zero-knowledge proof technique [35], as described in Figure 5. Finally, if the equation holds, the User is assured of possessing a legitimate identity, although the CSP would have no knowledge of that identity.

(iv) The process of authenticating the User identity certificate is described in Figure 6. If the equations hold, the User is assured of possessing a legitimate identity certificate, although the CSP would have no knowledge of that certificate.

(v) It must also be verified that the identity and the identity certificates belong to the same User. The verification process is described in Figure 7. If the equations are true, the identity is bound to the identity certificate.

(vi) Moreover, the legitimacy of the reputation and that the reputation belongs to the User must also be verified. The reputation is open to the public, so it is authenticated using the IdP public key. When verifying that the reputation belongs to the User, the User identity is hidden. To this end, a secret message is selected, and ownership of the reputation certificate can be verified only if the User has the secret

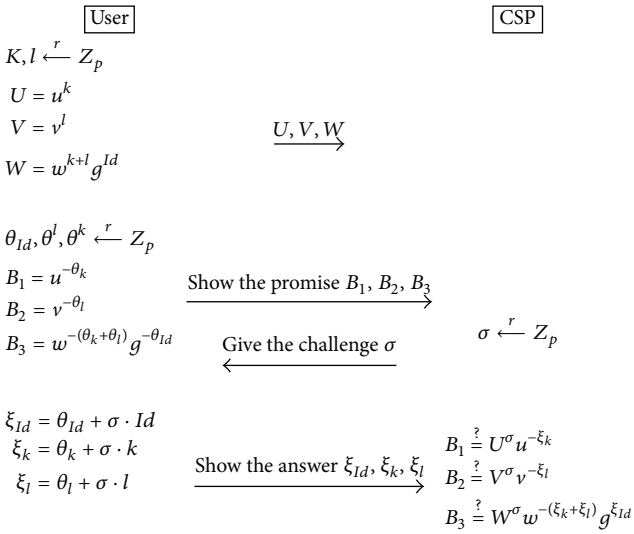


Figure 5

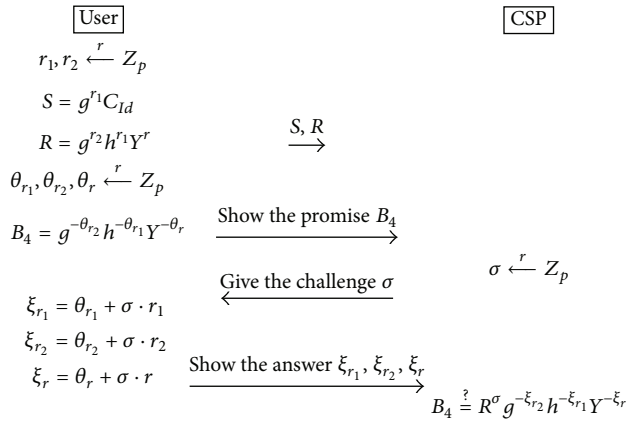


Figure 6

message, as described in Figure 8(a). Linkability between the reputation certificate and the User identity is then verified, as described in Figure 8(b). If the equations hold, the reputation and reputation certificate are concluded to belong to the User.

(vii) Finally, we verify that the encrypted identity certificate is from the IdP according to the encrypted identity.

We assume that

$$\begin{aligned} B_{11} &= e\left(g, XWR\right)^{\theta_{r_1}} e\left(S, w\right)^{\theta_k+\theta_l} e\left(g, w\right)^{-(\theta_{\delta_1}+\theta_{\delta_2})} \\ &\quad \cdot e\left(S, g\right)^{\theta_{r_2}} e\left(g, g\right)^{-\theta_{\delta_3}} e\left(S, h\right)^{\theta_{r_1}} e\left(g, h\right)^{-\theta_{\delta_4}}, \\ \xi &= e\left(g, XWR\right)^{\xi_{r_1}} e\left(S, w\right)^{(\xi_k+\xi_l)} e\left(g, w\right)^{-(\xi_{\delta_1}+\xi_{\delta_2})} \\ &\quad \cdot e\left(S, g\right)^{\xi_{r_2}} e\left(g, g\right)^{-\xi_{\delta_3}} e\left(S, h\right)^{\xi_{r_1}} e\left(g, h\right)^{-\xi_{\delta_4}} \\ &\quad \cdot \left(\left(\frac{e\left(g, g\right)}{e\left(S, XWR\right)}\right)^{\sigma}\right), \end{aligned} \tag{1}$$

where, if $B_{11} = \xi$, the abovementioned conclusion is verified because if the IdP private key is input, we obtain the output $e(C_{Id}, Xg^{Id}Y^r) = e(g, g)$.

Thus far, the legitimacy of the Id, C_{Id}, $C_{rep}C_{Id}$, and C_{rep} has been verified. Moreover, the linkability between the Id and the C_{Id} and C_{rep} has been verified. Though the Id has been encrypted, it can be verified that the C_{Id} derives from the IdP.

Together with the verification of the legitimacy of the reputation, all of the above comprise the operation $Authenticate(P_u, C_{rep})$. Because the reputation need not be secret, the reputation can be authenticated simply by inputting the IdP public key.

Next, we address the $e(C_{rep}, Xg^{rep}Y^q) = e(g, g)$ operation $Gen\_pny(Id, C_{Id}, rep)$. Here, the pseudonym P_u is just the random challenge σ used by the CSP for authenticating the User. P_u meets cloud computing security requirements. The security analysis will be conducted in the next section.

Our implementation of the Reputation Computation stage is described as follows:

(i) Reputation Withdrawal: the operation $Withdraw(Id)$ returns the reputation certificate, which is signed by the IdP. The specific implementation of the reputation certificate is equivalent to that of the identity certificate.

(ii) Reputation Backfeeding: we continue to use the short signature [38] to sign the feedback given by the CSP; that is, $C_{rep\_new} \leftarrow \langle g^{1/(i+rep\_new+jn)}, n\rangle$, where n is randomly selected: $n \xleftarrow{r} Z_p$.

(iii) Blinding: this operation is performed by the User and the IdP. The IdP possesses the key pair (M, m), where $M = g^{-m}$. The IdP randomly selects $r_b \xleftarrow{r} Z_p$ and computes $x_b = g^{r_b}$. Then, P_u obtains x_b from the IdP and selects a random number $Nonce$ that is to be signed. P_u computes $x_b^* = g^{u_b} M^{-d_b} x_b$, $e_b^* = H(x_b^*, Nonce)$, and $e_b = e_b^* + d_b$, where $d_b \xleftarrow{r} Z_p$, $u_b \xleftarrow{r} Z_p$. The User assigns e_b as the blind signature of $Nonce$; that is, $Nonce\_blind = e_b$.

(iv) Applying for Reputation Update. The IdP verifies the reputation feedback, which is signed by the CSP. If the equation $e(g^{1/(i+rep\_new+jn)}, Ig^{rep\_new}J^n) = e(g, g)$ holds, we conclude that the verification has passed.

(v) Blind Signing: the IdP computes $y_b = r_b + e_b \cdot m$ and $C_{blind} = y_b$. This is also the process of signing the blind random number $Nonce\_blind$. The IdP does not know the plaintext of $Nonce$ throughout the process.

(vi) Reputation Update Confirmation: the User removes the blindness of C_{blind} (i.e., y_b) through computing $y_b^* = y_b + u_b$. Then, the pair (e_b^*, y_b^*) is obtained, which is the signature of $Nonce$ denoted by C_{blind}. Finally, the User calls $Confirm\_Update(C_{Nonce}, Nonce)$ to confirm the update of the reputation. The IdP determines if the equation $x_b^* \stackrel{?}{=} g^{y_b^*} M^{e_b^*}$ holds. If so, the IdP then examines the equation $e_b^* \stackrel{?}{=} H(x_b^*, Nonce)$. When both equations hold, the IdP concludes that the signature is valid.

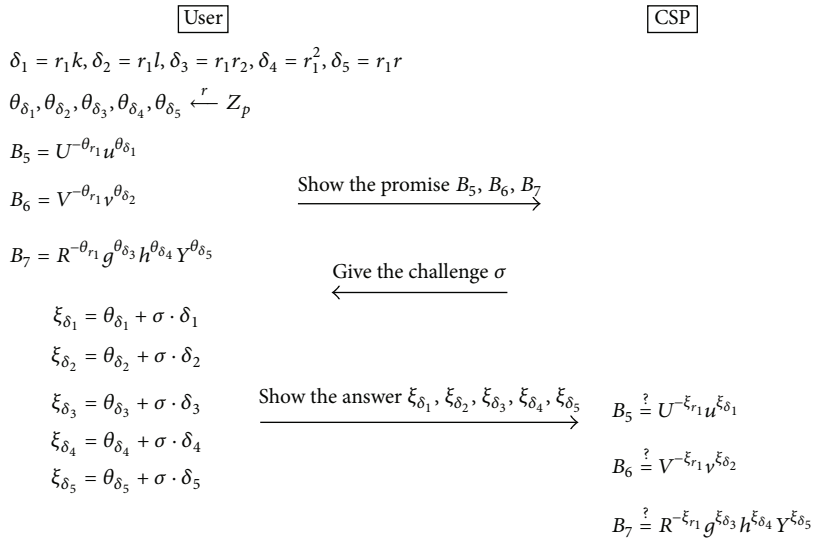


Figure 7

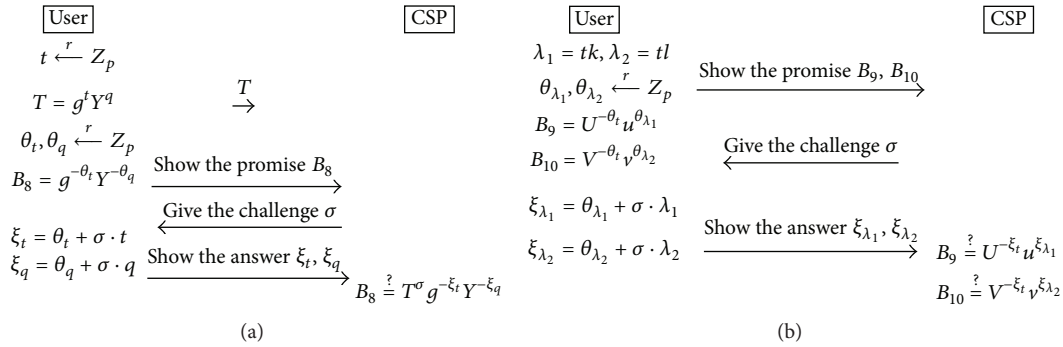


Figure 8

After verifying the blind signature, the IdP performs the actual reputation updating operation. We introduce the feedback based reputation calculation method [39] to estimate the User reputation. Here, three factors determine the total value of User trustworthiness, that is, rep_new, rep_h, rep_u, where rep_new represents the new reputation feedback the CSP provides after the transaction, rep_h is the history feedback provided by the previously accessed CSP, and rep_u is the overall reputation the User had prior to the current transaction. After accessing the service, the User reputation is calculated by $rep = (W\_new \times rep\_new) + (W\_h \times rep\_h) + (W\_u \times rep\_u)$. To regulate the value of the reputation, we let rep_new, rep_h, rep_u be normalized to values in $[0, 1]$. W_new, W_h, W_u denote the new feedback weight, the updated history feedback weight, and the updated overall reputation weight prior to the current transaction, respectively. All the weights have values in $[0, 1]$ and satisfy the constraint $W\_new + W\_h + W\_u = 1$. The details have been described in [39].

In the Pseudonym Disclosure stage, the DA and IdP decrypt the User pseudonym and then restore the User Id. The details of the two operations are as follows:

(i) Anonymity Removal: the DA decrypts the secret information (U, V, W) using the secret key (b, d) in the possession of the DA to restore g^{Id}; that is, $g^{Id} = W/(U^b V^d)$.

(ii) Identity Mapping: the IdP maps g^{Id} to the User Id using the internal mapping table that is built at the beginning.

Through the abovementioned operations, we can deliver an identity management service. The User generates a pseudonym using this service and then employs the pseudonym as the identity for accessing cloud services without exposing any real identity information. The User reputation, within a certain range, indicates to what extent the CSP can trust the user, based on the possibility that the User may carry out malicious activities.

5. Model Analysis

In this section, we verify the correctness and provide a detailed security analysis of the proposed model.

5.1. Correctness of the Model. The correctness of the model includes User Registration correctness, Reputation Calculation accuracy, Identity Authentication correctness, and Pseudonym Disclosure correctness.

Definition 1. User Registration correctness is described as follows. If the identity certificate and the reputation certificate can be verified with a probability of 1, then one concludes that the User Registration is correct. This is given by the following:

$$\Pr\left[\begin{array}{c}\left(PK_{Id\_s}, \mathrm{SK}_{Id\_s}\right) \longleftarrow Setup\_IdP\_Idsign\left(1^{\lambda}\right)\\ \left(C_{Id}, rep, C_{rep}\right) \longleftarrow Register\left(SK_{Id\_s}, Id\right)\\ CheckReg\left(PK_{Id\_s}, C_{Id}, rep, C_{rep}\right) = 1\end{array}\right] = 1. \tag{2a}$$

Theorem 2. *RIM User Registration is correct.*

Proof. If the above probability is 1, then the output of each operation must be correct, and the identity certificate and the reputation certificate can pass the verification using the public keys in the possession of the IdP. The validity of the identity certificate was verified in [38], and we need only to verify the validity of the reputation certificate. The following formula (3) confirms that $e(g^{1/(x+rep+yq)}, Xg^{rep}Y^{q}) = e(g, g)$, verifying the reputation certificate is valid. Consider

$$\begin{aligned} &e\left(g^{1/(x+rep+yq)}, Xg^{rep}Y^{q}\right)\\ &\quad= e\left(g^{1/(x+rep+yq)}, g^{x}g^{rep}\left(g^{y}\right)^{q}\right)\\ &\quad= e\left(g^{1/(x+rep+yq)}, g^{x+rep+yq}\right) = e\left(g, g\right). \end{aligned} \tag{3}$$

□

Definition 3. Identity Authentication correctness is described as follows. The CSP authenticates the User through the User pseudonym P_u. If all the operations return correct outputs with a probability of 1, we conclude that the Identity Authentication is correct. This is given by the following:

$$\Pr\left[\begin{array}{c}\left(PK_{Id\_s}, SK_{Id\_s}\right) \longleftarrow Setup\_IdP\_Idsign\left(1^{\lambda}\right)\\ \left(C_{Id}, rep, C_{rep}\right) \longleftarrow Register\left(SK_{Id\_s}, Id\right)\\ CheckReg\left(PK_{Id\_s}, C_{Id}, rep, C_{rep}\right) = 1\\ P_u \longleftarrow Gen\_Pny\left(Id, C_{Id}, rep\right)\\ Authenticate\left(P_u, C_{rep}\right) = 1\end{array}\right] = 1. \tag{2b}$$

Theorem 4. *RIM Identity Authentication is correct.*

Proof. The CSP must authenticate the User identity, the validity of the identity certificate, the linkability of the identity and identity certificate, the validity of the reputation certificate, and the linkability of the identity and reputation certificate. All of the above have been proven in [25] with the exception of the linkability of the *Id* and reputation certificate. Therefore, we only verify here the linkability of the *Id* and reputation certificate. The verification follows the same approach as that of formula (3) but is conducted by the CSP. Verifying the linkability of the *Id* and reputation certificate follows according to formula (4), which indicates that the User owns the reputation certificate, and formulas (5) and (6), which verify that the User identity is linked with reputation certificate. Consider

$$\begin{aligned} T^{\sigma}g^{-\xi_t}Y^{-\xi_q} &= \left(g^{t}Y^{q}\right)^{\sigma}g^{-(\theta_t+\sigma\cdot t)}Y^{-(\theta_q+\sigma\cdot q)}\\ &= g^{t\sigma}Y^{\sigma q}g^{-\sigma t}Y^{-\sigma q}g^{-\theta_t}Y^{-\theta_q} = g^{-\theta_t}Y^{-\theta_q}\\ &= B_8 \end{aligned} \tag{4}$$

$$\begin{aligned} U^{-\xi_t}u^{\xi_{\lambda_1}} &= U^{-(\theta_t+\sigma\cdot t)}u^{\theta_{\lambda_1}+\sigma\cdot\lambda_1} = U^{-\theta_t}u^{\theta_{\lambda_1}}U^{-\sigma\cdot t}u^{\sigma\cdot tk}\\ &= U^{-\theta_t}u^{\theta_{\lambda_1}}U^{-\sigma\cdot t}U^{\sigma\cdot t} = U^{-\theta_t}u^{\theta_{\lambda_1}} = B_9 \end{aligned} \tag{5}$$

$$\begin{aligned} V^{-\xi_t}v^{\xi_{\lambda_2}} &= V^{-(\theta_t+\sigma\cdot t)}v^{\theta_{\lambda_2}+\sigma\cdot\lambda_2} = V^{-\theta_t}v^{\theta_{\lambda_2}}V^{-\sigma\cdot t}v^{\sigma\cdot tl}\\ &= V^{-\theta_t}v^{\theta_{\lambda_2}}V^{-\sigma\cdot t}V^{\sigma\cdot t} = V^{-\theta_t}v^{\theta_{\lambda_2}} = B_{10}. \end{aligned} \tag{6}$$

□

The Reputation Computation accuracy involves two aspects: (1) the correctness of the operations such as reputation acquisition, reputation feedback, and reputation update; and (2) the fact that the reputation evaluation algorithm can accurately calculate the user degree of trustworthiness. The accuracy of the evaluation algorithm has been verified in [39]. Therefore, in this paper, we only define the correctness of the operations.

Definition 5. Reputation Calculation correctness is described as follows. If the reputation-related operations return correct outputs with a probability of 1, one concludes that the Reputation Calculation is correct. This is given by the following:

$$\Pr\left[\begin{array}{c}\left(PK_{Id\_s}, SK_{Id\_s}\right) \longleftarrow Setup\_IdP\_Idsign\left(1^{\lambda}\right)\\ \left(PK_{Id\_b}, SK_{Id\_b}\right) \longleftarrow Setup\_IdP\_blind\left(1^{\lambda}\right)\\ \left(PK_{CSP}, SK_{CSP}\right) \longleftarrow Setup\_CSP\left(1^{\lambda}\right)\\ \left(rep, C_{rep}\right) \longleftarrow Withdraw\left(Id\right)\\ CheckReg\left(PK_{Id\_s}, C_{Id}, rep, C_{rep}\right) = 1\\ P_u \longleftarrow Gen\_Pny\left(Id, C_{Id}, rep\right)\\ Authenticate\left(P_u, C_{rep}\right) = 1\\ \left(rep\_new, C_{rep\_new}\right) \longleftarrow Grant\left(P_u\right)\\ Nonce\_blind \longleftarrow Blind\left(Nonce\right)\\ Update\left(rep\_new, C_{rep\_new}, Nonce\_blind\right) = 1\\ C_{blind} \longleftarrow Blind\_sign\left(Nonce\_blind\right)\\ Confirm\_Update\left(C_{blind}\right) = 1\end{array}\right] = 1. \tag{2c}$$

Theorem 6. *RIM Reputation Calculation operations are correct.*

Proof. The operation $Withdraw(Id)$ and the subsequent verification steps of the reputation are equivalent to the registration operations. The verification of the reputation feedback is similar to that given in formula (3). Because the correctness of the registration operations has been proven in Theorems 2 and 4, we here prove only the correctness of the blind signature. Formula (7) removes the blindness of the signature, obtaining the intermediate value x_b^*. If the equation $e_b^* \stackrel{?}{=} H(x_b^*, Nonce)$ holds, we conclude that the blind signature is correct. Along with Theorems 2 and 4, the proof verifies the correctness of the reputation operations. One has

$$\begin{aligned} g^{y_b^*} M^{e_b^*} &= g^{y_b+u_b} M^{e_b^*} = g^{r_b+e_b\cdot m+u_b} M^{e_b-d_b} \\ &= x_b g^{e_b\cdot m} g^{u_b} g^{-m(e_b-d_b)} \\ &= x_b g^{e_b\cdot m} g^{u_b} g^{-m\cdot e_b} g^{-m(-d_b)} = x_b g^{u_b} M^{-d_b} \\ &= x_b^*. \end{aligned} \tag{7}$$

□

Definition 7. Pseudonym Disclosure correctness is described as follows. The DA decrypts the pseudonym to restore the User Id with a probability of 1. This is given by the following:

$$\Pr\left[\begin{array}{c} (PK_{Id\_s}, SK_{Id\_s}) \longleftarrow Setup\_IdP\_Idsign\left(1^\lambda\right) \\ (PK_{DA}, SK_{DA}) \longleftarrow Setup\_DA\left(1^\lambda\right) \\ \left(rep, C_{rep}\right) \longleftarrow Withdraw\,(Id) \\ CheckReg\left(PK_{Id\_s}, C_{Id}, rep, C_{rep}\right) = 1 \\ P_u \longleftarrow Gen\_Pny\,(Id, C_{Id}, rep) \\ Authenticate\left(P_u, C_{rep}\right) = 1 \\ g^{Id} \longleftarrow De\_Anonymity\,(P_u, SK_{DA}) \\ Id \longleftarrow Map\left(g^{Id}\right) \end{array}\right] = 1. \tag{2d}$$

Theorem 8. *RIM Pseudonym Disclosure is correct.*

Proof. Operations conducted prior to the Pseudonym Disclosure process have been proven in Theorems 2, 4, and 6. The correctness of decrypt (U, V, W) has been proven in [36]. Finally, the IdP queries the mapping table to obtain the User Id. Therefore, we conclude that the Pseudonym Disclosure is correct. □

5.2. Security of the Model

5.2.1. Security of Reputation. Before discussing the security of reputation, we assume that the private keys of the IdP, CSP, and User have not been leaked. If the private keys were leaked, the RIM would be open to the public. The data of the IdP are stored in a cloud environment, and the cloud service is open to the public. Therefore, an attacker can obtain an Id and reputation pair, thereby conducting a plaintext attack by analyzing them. In this paper, we will not consider a situation where an attacker modifies the reputation through the cloud underlying infrastructure, and the attacks are all chosen as plaintext attacks.

Unforgeability of reputation refers to the fact that an unauthorized user has no way of modifying their own or another's reputation value. If the attacker (or Adversary; Adv) wishes to modify a reputation value without permission, the Adv must forge a reputation certificate issued by the IdP or a feedback certificate signed by the CSP. Next, we provide a formal definition of reputation unforgeability.

Definition 9. If a probabilistic polynomial-time adversary Adv has the advantage $Advantage_{unf}(\lambda) = \Pr[(PrivK_{unf}(\lambda) = 1) - 1/2]$, which is negligible in λ, then one concludes that the reputation value is unforgeable. The experiment $PrivK_{unf}(\lambda)$ is defined as follows:

(i) Adv is given the public parameters $\langle p, g, G, G_T, e\rangle$, as described in the Environment Initialization stage. The register random oracle O_{reg} provides Adv with a User identity, reputation value, and reputation certificate. The feedback random oracle O_{CSP} provides Adv with reputation feedback and the feedback certificate. The pseudonym random oracle O_{Id} provides Adv with the User pseudonym P_u.

(ii) The Adv generates $(Id_0, rep_0), (Id_1, rep_1)$ and $((P_u)_0, rep\_new_0), ((P_u)_1, rep\_new_1)$.

(iii) The challenger randomly selects b from $\{0, 1\}$; that is, $b \leftarrow \{0, 1\}$. Then, the challenger calls $(rep_b, (C_{rep})_b) \leftarrow Withdraw(Id_b)$ and $(rep\_new_b, (C_{rep\_new})_b) \leftarrow Grant((P_u)_b)$ to acquire the pair $(rep_b, (C_{rep})_b)$ and $(rep\_new_b, (C_{rep\_new})_b)$.

(iv) The Adv gives a guess b', and if $b = b'$, then output is 1; otherwise, output is 0.

Now, we reduce this Adv to an adversary $(Adv)_b$ for the BB signature. Firstly, $(Adv)_b$ initializes a hash table H for the simulation of the random oracles O_{reg} and O_{CSP} that are employed by Adv. If Adv queries (Id, rep) or (P_u, rep_new), $(Adv)_b$ would resort to H. If the query has already been on the table, $(Adv)_b$ would return the corresponding value; if not, $(Adv)_b$ would sample a (Id, rep) or (P_u, rep_new) and place it in H and return it in the end. When Adv queries a reputation signature included in (Id, rep) or (P_u, rep_new), $(Adv)_b$ forwards the query to BB-signing oracle O_{BB} and returns the response. After sufficient queries, Adv challenges the challenger using $(Id_0, rep_0), (Id_1, rep_1)$, or $((P_u)_0, rep\_new_0)$, $((P_u)_1, rep\_new_1)$. $(Adv)_b$ extracts the pair (rep_0, rep_1) or (rep_new_0, rep_new_1) and challenges the BB-signing challenger. The BB-signing challenger picks a number b in $\{0, 1\}$. $(Adv)_b$ then returns $(C_{rep})_b$ or $(C_{rep\_new})_b$ to Adv. Adv makes a guess of b' and gives it to the BB-signing challenger through $(Adv)_b$. If $b = b'$, then not only does Adv have the ability to forge a reputation certificate but also $(Adv)_b$ can forge a

BB signature. According to the above mentioned method, we reduce Adv to $(Adv)_b$, and we conclude that if the BB signature is secure, then the reputation certificate cannot be forged.

Verifiable Reputation means that the reputation indeed belongs to the authenticated user. If the adversary can forge a pseudonym and the pseudonym can pass verification by the CSP while the DA cannot open the pseudonym or only open it as an unregistered identity, then we conclude that the adversary is successful. This property is guaranteed by Hidden Identity-Based (HIB) Signatures [25].

Definition 10. If a probabilistic polynomial-time adversary Adv has the advantage $Advantage_{ver}(\lambda) = \Pr[(PrivK_{ver}(\lambda) = 1) - 1/2]$, which is negligible in λ, then one concludes that the reputation value is verifiable. The experiment $PrivK_{ver}(\lambda)$ is defined as follows:

(i) The register random oracle O_{reg} provides Adv with a User identity, reputation value, and reputation certificate, and the pseudonym random oracle O_{Id} provides Adv with a User pseudonym P_u. Adv obtains the public parameters $\langle p, g, G, G_T, e\rangle$ as described in the Environment Initialization stage.

(ii) Adv generates the pair $(Id_0, (C_{Id})_0, rep_0)$ and $(Id_1, (C_{Id})_1, rep_1)$ and then gives it to the challenger.

(iii) The challenger randomly selects b from $\{0, 1\}$; that is, $b \leftarrow \{0, 1\}$. Then, he calls $(P_u)_b \leftarrow Gen\_Pny(Id_b, (C_{Id})_b, rep_b)$ to acquire the pseudonym $(P_u)_b$.

(iv) Adv gives a guess of b', and if $b = b'$, $1 \leftarrow Authenticate((P_u)_b, (C_{rep})_b)$ and $\perp \leftarrow De\_Anonymity((P_u)_b, SK_{DA})$ hold, and then output is 1; otherwise, output is 0.

Now, we assume that an adversary $(Adv)_H$ is against the HIB signature. Using three random oracles, namely, the registration random oracle $RegOracle(Id)$, the User identity random oracle $CorruptOracle(Id)$, and the signature oracle $SignOracle(Id, cert_{id}, m)$, $(Adv)_H$ acquires the User identity, identity certificate, and the signature of a message signed by the User. Moreover, these three random oracles can simulate O_{reg} and O_{Id}. In [36], the random oracle $CorruptOAOracle()$ was used for disclosing a pseudonym to perform a cipher text attack. However, in this paper, we assume that the DA is trustworthy. Therefore, we simulate only a plaintext attack. If Adv can disclose a pseudonym, then $(Adv)_H$ must be able to break the HIB signature. Now, we begin the process of reducing Adv to $(Adv)_H$. $(Adv)_H$ removes the part relevant to reputation from $(Id_0, (C_{Id})_0, rep_0)$ and $(Id_1, (C_{Id})_1, rep_1)$ provided by Adv and then proceeds with the HIB signature process. During the authentication process, we maintain a constant challenge factor σ. The reputation was included when σ was generated. By analyzing the Σ-protocol, we know that σ is randomly selected, so its value does not affect the final result of authentication. Finally, $(Adv)_H$ takes σ as a pseudonym $(P_u)_b$ and returns it to Adv. $(P_u)_b$ contains the information regarding a User identity. Adv gives a guess of b', and if Adv can successfully forge a pseudonym, then $(Adv)_H$ can crack the HIB signature with the same probability.

Nonrepudiation means that, under any circumstances, the User must admit that the reputation that has been submitted belongs only to the User, and, therefore, regardless of how low the reputation value, the User can never deny ownership. If the adversary has the ability to forge a pseudonym and the DA opens it to find that it corresponds to a legitimate registered user identity, then we conclude that the adversary is successful.

Definition 11. If a probabilistic polynomial-time adversary Adv has the advantage $Advantage_{nre}(\lambda) = \Pr[(PrivK_{nre}(\lambda) = 1) - 1/2]$, which is negligible in λ, then one concludes that the reputation value has the property of Nonrepudiation. The experiment $PrivK_{nre}(\lambda)$ is defined as follows:

(i) Adv is given the public parameters $\langle p, g, G, G_T, e\rangle$ as described in the Environment Initialization stage. The register random oracle O_{reg} provides Adv with a User identity, reputation value, and reputation certificate. The pseudonym random oracle O_{Id} provides Adv with a User pseudonym P_u.

(ii) Adv generates $(Id_0, (C_{Id})_0, rep_0)$ and $(Id_1, (C_{Id})_1, rep_1)$ and then gives it to the challenger.

(iii) The challenger randomly selects b from $\{0, 1\}$; that is, $b \leftarrow \{0, 1\}$. Then, the challenger calls $(P_u)_b \leftarrow Gen\_Pny(Id_b, (C_{Id})_b, rep_b)$ to acquire the pseudonym $(P_u)_b$.

(iv) Adv gives a guess b', and if $b = b'$, $1 \leftarrow Authenticate((P_u)_b, (C_{rep})_b)$ and $1 \leftarrow De\_Anonymity((P_u)_b, SK_{DA})$ are true, and then output is 1; otherwise, output is 0.

The method that reduces Adv to an adversary for the HIB signature is equivalent to the method used in the analysis of Verifiable Reputation. We therefore omit the reducing process.

5.2.2. Secure Anonymity. In addition to the aforementioned reputation security, it is necessary to consider the security of anonymity. To this end, we assume that the IdP is not safe; namely, its private keys can be leaked. Also, we assume that the IdP and CSP can collude together to compromise anonymity. These assumptions comply with the actual situation, for the IdP and CSP may belong to the same institution. However, in the Reputation Calculation process, we assume that private keys in the possession of the IdP cannot be leaked to the User but can be leaked to the CSP because reputation is used to constrain the User, and the IdP and User are antithetical.

Anonymity security includes the unlinkability not only between pseudonyms (denoted as P_u-unlinkability) but also between a pseudonym and a User Id (denoted as Id-P_u-unlinkability). P_u-unlinkability includes the unlinkability between the pseudonyms of the same user and the unlinkability between the pseudonyms of different users.

With regard to the probability $\Pr[(P_u)_0 = (P_u)_1]$, if the two pseudonyms $(P_u)_0$ and $(P_u)_1$ given in Definition 12 are negligible, then we conclude that $(P_u)_0$ and $(P_u)_1$ are unlinkable.

A number of factors make one pseudonym different from another. Firstly, for different users, their identities are different, and the randomly selected parameter r ensures that the identity certificates are different. Moreover, the parameters used for encrypting the identity and identity certificate are different, making S, R, U, V, and W different. The parameters used for authentication are also different. Finally, the reputation of each user is different in most cases. The above factors constitute the challenge factor σ (i.e., P_u). If an association exists between pseudonyms, the one-way hash function used to generate pseudonyms will be compromised, although this is impossible. Secondly, for the pseudonyms of the same user, the parameters used to generate σ, excluding the identities, are different. Therefore, in this case, the pseudonyms are unlinkable.

Id-P_u-unlinkability ensures that a user identity cannot be inferred from the user pseudonym. In the process of user authentication, U, V, W, S, and R, which are the cipher text of a user identity and identity certificate, generate the pseudonym P_u along with other factors. The unlinkability between P_u and the user identity is guaranteed by the strength of linear encryption.

Definition 12. If a probabilistic polynomial-time adversary Adv has the advantage $Advantage_{IPU}(\lambda) = \Pr[(PrivK_{IPU}(\lambda) = 1) - 1/2]$, which is negligible in λ, then we conclude that a User identity and corresponding pseudonym are unlinkable. The experiment $PrivK_{IPU}(\lambda)$ is defined as follows:

(i) Adv is given the public parameters $\langle p, g, G, G_T, e\rangle$, as described in the Environment Initialization stage. The register random oracle O_{reg} provides Adv with a User identity, reputation value, and reputation certificate. The pseudonym random oracle O_{Id} provides Adv with a User pseudonym P_u.

(ii) Adv generates $(Id_0, (C_{Id})_0, rep_0)$ and $(Id_1, (C_{Id})_1, rep_1)$ and then gives them to the challenger.

(iii) The challenger randomly selects b from $\{0, 1\}$; that is, $b \leftarrow \{0, 1\}$. Then, the challenger calls $(P_u)_b \leftarrow Gen\_Pny(Id_b, (C_{Id})_b, rep_b)$ to acquire the pseudonym $(P_u)_b$.

(iv) Adv gives a guess of b', and if $b = b'$ then output is 1; otherwise, output is 0.

Nonrepudiation and Verifiable Reputation are based on the unlinkability between a User identity and pseudonym. Therefore, in the analysis of nonrepudiation and Verifiable Reputation, we have proved that a User identity and pseudonym are unlinkable.

5.2.3. Reputation and Anonymity. The introduction of reputation will influence User anonymity. The CSP can obtain User identity information through reputation in three ways: (1) reputation feedback; (2) reputation submitted by P_u; and (3) changes of reputation. Subsequent analysis indicates that all three ways will fail, provided the blind signature is not compromised, greater than 50 users are registered in the IdP, and the CSP is not synchronized with the IdP.

The introduction of the blind signature ensures that the process of updating a reputation will not disclose User identity information. P_u submits the *Nonce_blind* to the IdP to apply for modification of the reputation value. The User (the generator of P_u) submits the signature of *Nonce* to the IdP to confirm the reputation update. The technology of blind signature ensures the unlinkability between both signatures of *Nonce_blind* and *Nonce*. Therefore, even collusion between the IdP and CSP cannot reveal User identity information.

When the number of Users registered in the IdP is greater than 50, the CSP cannot locate a User through their reputation value. We assume that the number of users registered in the IdP is n and divide the interval of reputations, say $[0, 1]$, into N independent values from which these n users choose reputations independently (User reputation values are calculated independently of each other, and the reputation values can be considered uniformly distributed on the interval.). The probability distribution of two users with an equivalent reputation is then $P(n) \sim 1 - 1/\exp(n^2/2N)$ (http://en.wikipedia.org/wiki/birthday_Problem). As we can see from $P(n)$, when $N = 100$, the probability that two users have equivalent reputation values increases rapidly with increasing n. When n approaches 50, the probability resides very close to 1. When $N = 300$, as the number of users approaches 60, the probability continues to reside very close to 1. For simplicity, we take $N = 100$ in this paper. Therefore, when n is greater than 50, a set of users will have equivalent reputations, making it difficult for the CSP to locate a User based upon their reputation. To make the model more general, we assume that A is the collection of users that have the same reputation; that is, $A = \{a_1, a_2, \ldots, a_m\}$ for $1 \le m \le n$ (at least one user in the collection provides a reputation). To mine user privacy, the CSP must continuously track users. Assuming the CSP seeks to track user a_j for $j \in \{1, \ldots, m\}$, after the next transaction between a_j and the CSP, the CSP will acquire a set of users B based on the reputation provided by the User through the corresponding pseudonym (P_u). The CSP will obtain the solution $A \cap B = \varnothing, a_j \notin (A \cap B)$, or $a_j \in (A \cap B)$ regardless of whether or not the reputation is identical in these two steps. Therefore, when $n > 50$, the CSP cannot obtain User identity information from the reputation value. The openness of cloud computing ensures that n will be far greater than 50, so that, in practice, the reputation will not disclose User identities.

Through disrupting the synchronization of the IdP and CSP, the CSP cannot locate a User through an update of the reputation value. For the aforementioned user set A, when the CSP gives feedback to a pseudonym, the CSP would monitor the reputation changes in the IdP and then link the pseudonym to a user identity. In this case, the CSP must be synchronized with the IdP. For example, suppose a pseudonym $(P_u)_j$ of user a_j accesses the CSP with reputation

rep_j. After that, the CSP grants feedback to $(P_u)_j$ to increase the reputation value. Then, the CSP monitors the users in set A whose reputation has been increased, links $(P_u)_j$ to a_j, and then records the operations of $(P_u)_j$. With a long-term monitor, the CSP will eventually record all the actions of the user and compromise the user's privacy. To solve this problem, the User can choose a random waiting time to confirm the reputation value update so that the synchronization will be disrupted; thus, the CSP cannot determine if the pseudonym that the CSP has just granted belongs to the user whose reputation value has changed.

6. Conclusions

In this paper, we designed and implemented RIM, an identity management model for cloud computing that enables users to access cloud services using pseudonyms so as to ensure the unlinkability not only between different pseudonyms but also between a user and their corresponding pseudonym. In this way, user privacy can be protected. In addition, by calculating the reputation of users, RIM can assist CSPs to identify malicious users. RIM compensates for the shortcomings of identity management introduced by the multitenant feature and openness of the cloud computing environment.

In this paper, we assumed that the CSP honestly provides credible reputation feedback to users. However, in fact, malicious CSPs that provide dishonest assessments of user behavior may exist. In addition, a CSP may violate service level agreements (SLA). The various security vulnerabilities of CSPs pose a threat to users. Therefore, our future goal is to assess the credibility of CSPs and improve the reputation evaluation mechanism so as to provide better protection of user privacy. The access control mechanism in the cloud environment by means of reputation is also a valuable research topic that can be explored in the future.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] P. Mell and T. Grance, *The NIST Definition of Cloud Computing*, National Institute of Standards and Technology, Gaithersburg, Md, USA, 2011.

[2] P. Mell and T. Grance, "The NIST definition of cloud computing," Special Publication 800-145, National Institute of Standards and Technology, 2011.

[3] E. Olden, "Architecting a cloud-scale identity fabric," *Computer*, vol. 44, no. 3, pp. 52–59, 2011.

[4] E. Bertino, F. Paci, and R. Ferrini, "Privacy-preserving digital identity management for cloud computing," *IEEE Data Engineering Bulletin*, vol. 32, no. 1, pp. P21–P27, 2009.

[5] P. Angin, B. Bhargava, R. Ranchal et al., "An entity-centric approach for privacy and identity management in cloud computing," in *Proceedings of the 29th IEEE Symposium on Reliable Distributed Systems (SRDS '10)*, pp. 177–183, New Delhi, India, November 2010.

[6] S. Ruj, M. Stojmenovic, and A. Nayak, "Privacy preserving access control with authentication for securing data in clouds," in *Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid '12)*, pp. 556–563, IEEE, Ottawa, Canada, May 2012.

[7] S. S. M. Chow, Y.-J. He, L. C. K. Hui, and S. M. Yiu, "SPICE—simple privacy-preserving identity-management for cloud environment," in *Applied Cryptography and Network Security: Proceedings of the 10th International Conference, ACNS 2012, Singapore, June 26–29, 2012*, vol. 7341 of *Lecture Notes in Computer Science*, pp. 526–543, Springer, Berlin, Germany, 2012.

[8] J. Xiong, Z. Yao, J. Ma, X. Liu, Q. Li, and T. Zhang, "PRAM: privacy preserving access management scheme in cloud services," in *Proceedings of the International Workshop on Security in Cloud Computing (Cloud Computing '13)*, pp. 41–46, ACM, Hangzhou, China, May 2013.

[9] K. Ren, W. Lou, K. Kim, and R. Deng, "A novel privacy preserving authentication and access control scheme for pervasive computing environments," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 4, pp. 1373–1384, 2006.

[10] K. Govinda and P. Ravitheja, "Identity anonymization and secure data storage using group signature in private cloud," in *Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI '12)*, pp. 129–132, ACM, ind, August 2012.

[11] T. H. Noor, Q. Z. Sheng, S. Zeadally, and J. Yu, "Trust management of services in cloud environments: obstacles and solutions," *ACM Computing Surveys*, vol. 46, no. 1, article 12, 2013.

[12] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in P2P networks," in *Proceedings of the 12th International Conference on World Wide Web (WWW '03)*, pp. 640–651, ACM, May 2003.

[13] L. Xiong and L. Liu, "PeerTrust: a trust mechanism for an open peer-to-peer information system," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, 2004.

[14] R. Zhou and K. Hwang, "PowerTrust: a robust and scalable reputation system for trusted peer-to-peer computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 4, pp. 460–473, 2007.

[15] N. K. Sharma, V. Gaur, and S. K. Muttoo, "A dynamic reputation system with built-in attack resilience to safeguard buyers in e-market," *ACM SIGSOFT Software Engineering Notes*, vol. 37, no. 4, pp. 1–19, 2012.

[16] I. Pranata, R. Athauda, and G. Skinner, "Modeling decentralized reputation-based trust for initial transactions in digital environments," *ACM Transactions on Internet Technology*, vol. 12, no. 3, article 8, 35 pages, 2013.

[17] R. Shaikh and M. Sasikumar, "Trust framework for calculating security strength of a cloud service," in *Proceedings of the International Conference on Communication, Information and Computing Technology (ICCICT' 12)*, October 2012.

[18] T. H. Noor and Q. Z. Sheng, "Trust as a service: a framework for trust management in cloud environments," in *Web Information System Engineering—WISE 2011*, vol. 6997 of *Lecture Notes in Computer Science*, pp. 314–321, Springer, Berlin, Germany, 2011.

[19] F. G. Mármol, J. Girao, and G. M. Pérez, "TRIMS, a privacy-aware trust and reputation model for identity management systems," *Computer Networks*, vol. 54, no. 16, pp. 2899–2912, 2010.

[20] A. Post, V. Shah, and A. Mislove, "Bazaar: strengthening user reputations in online marketplaces," in *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation (NSDI '11)*, pp. 183–196, 2011.

[21] L.-H. Vu and K. Aberer, "Effective usage of computational trust models in rational environments," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 6, no. 4, article 24, 25 pages, 2011.

[22] E. Androulaki, S. G. Choi, S. M. Bellovin, and T. Malkin, "Reputation systems for anonymous networks," in *Privacy Enhancing Technologies*, vol. 5134 of *Lecture Notes in Computer Science*, pp. 202–218, Springer, Berlin, Germany, 2008.

[23] H. Rifà-Pous, "Anonymous reputation based reservations in e-commerce (AMNESIC)," in *Proceedings of the 13th International Conference on Electronic Commerce (ICEC '11)*, ACM, August 2011.

[24] M. H. Au and A. Kapadia, "PERM: practical reputation-based blacklisting without TTPs," in *Proceedings of the ACM Conference on Computer and Communications Security*, pp. 929–940, ACM, October 2012.

[25] T. Okamoto, "Provably secure and practical identification schemes and corresponding signature schemes," in *Advances in Cryptology—CRYPTO'92*, vol. 740 of *Lecture Notes in Computer Science*, pp. 31–53, Springer, Berlin, Germany, 1993.

[26] J.-Z. Luo, J.-H. Jin, A.-B. Song, and F. Dong, "Cloud computing: architecture and key technologies," *Journal on Communications*, vol. 32, no. 7, pp. 3–21, 2011.

[27] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1–11, 2011.

[28] A. Josang and S. Pope, "User centric identity management," in *Proceedings of the AusCERT Conference*, 2005.

[29] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*, vol. 196 of *Lecture Notes in Computer Science*, pp. 47–53, Springer, Berlin, Germany, 1985.

[30] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology—CRYPTO 2001*, vol. 2139 of *Lecture Notes in Computer Science*, pp. 213–229, Springer, Berlin, Germany, 2001.

[31] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *Advances in Cryptology—ASIACRYPT 2001*, vol. 2248 of *Lecture Notes in Computer Science*, pp. 514–532, Springer, Berlin, Germany, 2001.

[32] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology: Proceedings of Crypto 82*, pp. 199–203, Springer, New York, NY, USA, 1983.

[33] U. Feige, A. Fiat, and A. Shamir, "Zero-knowledge proofs of identity," *Journal of Cryptology*, vol. 1, no. 2, pp. 77–94, 1988.

[34] L. C. Guillou and J.-J. Quisquater, "A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory," in *Advances in Cryptology—EUROCRYPT '88*, vol. 330 of *Lecture Notes in Computer Science*, pp. 123–128, Springer, Berlin, Germany, 1988.

[35] C. P. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, vol. 4, no. 3, pp. 161–174, 1991.

[36] A. Kiayias and H. Zhou, "Hidden identity-based signatures," in *Financial Cryptography and Data Security*, vol. 4886 of *Lecture Notes in Computer Science*, pp. 134–147, Springer, Berlin, Germany, 2007.

[37] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Advances in Cryptology—CRYPTO 2004*, vol. 3152 of *Lecture Notes in Computer Science*, pp. 41–55, Springer, Berlin, Germany, 2004.

[38] D. Boneh and X. Boyen, "Short signatures without random oracles," in *Advances in Cryptology—EUROCRYPT 2004*, vol. 3027 of *Lecture Notes in Computer Science*, pp. 56–73, Springer, Berlin, Germany, 2004.

[39] B.-X. Li, L.-F. Wu, Z.-J. Zhou, and H.-B. Li, "Design and implementation of trust-based identity management model for cloud computing," *Computer Science*, vol. 41, no. 10, pp. 3-21.144–3-21.148, 2014.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 916418, 12 pages
http://dx.doi.org/10.1155/2015/916418

Research Article

Research on Cloud Computing Resources Provisioning Based on Reinforcement Learning

Zhiping Peng,[1] Delong Cui,[1] Jinglong Zuo,[1] and Weiwei Lin[2]

[1]*College of Computer and Electronic Information, Guangdong University of Petrochemical Technology, Maoming 525000, China*
[2]*School of Computer Science and Engineering, South China University of Technology, Guangzhou, China*

Correspondence should be addressed to Delong Cui; delongcui@163.com

Received 27 March 2015; Revised 2 June 2015; Accepted 4 June 2015

Academic Editor: Bao Rong Chang

As one of the core issues for cloud computing, resource management adopts virtualization technology to shield the underlying resource heterogeneity and complexity which makes the massive distributed resources form a unified giant resource pool. It can achieve efficient resource provisioning by using the rational implementing resource management methods and techniques. Therefore, how to manage cloud computing resources effectively becomes a challenging research topic. By analyzing the executing progress of a user job in the cloud computing environment, we proposed a novel resource provisioning scheme based on the reinforcement learning and queuing theory in this study. With the introduction of the concepts of Segmentation Service Level Agreement (SSLA) and Utilization Unit Time Cost (UUTC), we viewed the resource provisioning problem in cloud computing as a sequential decision issue, and then we designed a novel optimization object function and employed reinforcement learning to solve it. Experiment results not only demonstrated the effectiveness of the proposed scheme, but also proved to outperform the common methods of resource utilization rate in terms of SLA collision avoidance and user costs.

1. Introduction

The concept of cloud computing vividly reflects the characteristics of information service in Internet age; meanwhile the pursuit of the vision of cloud computing also brings new challenges to information technology. Acting as a significant application research, the data center is pushing a series of technology innovations to perform the key features of cloud computing, such as on-demand service, elasticity of extension, and massive data storage. The data center widely adopts virtualization technology to achieve the uncoupled mode of physical resource and application. Applications use Virtual Machine (VM) as a package unit to share various physical resources with others. Hence the resource schedule entities are represented by fine-grain VMs instead of coarse grain service machines. Virtualization technology provides convenience for the data center, but the VMs resource provisioning brings more challenges to the efficient management of data center infrastructure.

As one of the core issues for cloud computing, resource management aims to shield the underlying resource heterogeneity and complexity by adopting virtualization technology, which makes the massive distributed resources form a unified giant resource pool. Therefore, it can guarantee the efficient resource provision and use by implementing resources management methods and techniques rationally. Therefore, how to achieve effective management of cloud computing resources is faced with a number of new challenges, which are mainly shown in three types of imbalance.

(i) First, Imbalance in the Needs of Applications. Cloud computing application contains various behaviors of workload, from the control-intensive applications (such as search, sort, and analysis) to the data-intensive ones (image processing, simulation, modeling, data mining, etc.). In addition, it also includes the computationally intensive applications (iterative method, numerical method, financial modeling, etc.). The throughput of various applications depends heavily on

the VM resource provisioning, while not even any configuration can make all types of workloads run with optimal efficiency. Furthermore, most applications are featured by multiple types of workload. For example, control-intensive applications require more CPU resources for branch prediction, while data-intensive ones require more memory resources to avoid the frequent operation of reading and writing. The multitenant environment of cloud computing allows heterogeneous applications to share data center resources pool, and the demand for resources for each application is diverse, thus resulting in difficulty measuring the server-loading efficiency. Even for the different resources of the same server, it is prone to cause an imbalance, affecting the resource use efficiency. In cloud computing scenario, while simply the demand for forecasting and the rational purchase cannot solve this problem, a plausible resource provisioning scheme needs to be put forward to further solve the new contradiction between heterogeneous application and the unified resources-sharing pool.

(ii) Second, Imbalance in the Application Time. In reality, the server utilization of data center reaches merely 5% to 20%, while the peak workloads of many services are 2–10 times higher than the average. In addition to the different service loads in various periods of a day, most of the services vary in load demands according to the seasonal or other periodic changes (e.g., peak in December before Christmas sales and in photo processing sites after holiday); meanwhile some unexpected events (such as news) lead to changes. Few users deploy less resource than the peak demands, which is prone to waste the resources at the nonpeak time. As a result, the stronger the load is fluctuating, the more resources the users are wasting. In the cloud computing environment, this issue cannot be handled by static configuration mode. At the same time, the VM in the cloud is characterized by isolation performance, but mutual interference resulting from the resource competition between VMs cannot be avoided in the actual operation of the system process, affecting the performance of the whole cloud computing system.

(iii) Thirdly, Imbalance in the Distribution of Applications. For load balancing, the node servers of the load equalizers are not fixed physical machines but are VMs of the cloud, which require the load balancer to be equipped with the ability of dynamically adjusting the server cluster to the current user access, so as to avoid the resource waste and the situation that the current resources cannot meet the user's requests [1]. To some extent, numerous academic researches have been conducted on the trend prediction of the load and elastic assignment of resources; however, there are still some disadvantages. Having conducted a comprehensive study of academia and industry current status, we find that the research in this field has the following problems: first, it lacks flexibility. Apart from not considering the dynamic deployment of resources from a service-oriented perspective, some characteristics are not fully reflected, such as the elasticity of cloud computing and the characteristic that resources are adjusted to the user's needs. Second, it does not support the trend prediction, and the resource allocations are apparently lagging behind, affecting the user's experience and even being unable to meet some of the user requests at times.

In conclusion, the data center must solve the problems such as resource multiplexing, correlation, and dynamic management. High-efficiency dynamic management for virtualization resources is the core issue of the optimal resource scheduler and also the key of how various resource service systems eventually provide appropriate and satisfying resources for users.

Focusing on the accurate scaled cloud computing environment and efficient resources allocation under Service Level Agreement (SLA) and user cost constraints, we introduce two concepts, Segmentation SLA (SSLA) and Utilization Unit Time Cost (UUTC), and then propose an optimization resource provisioning scheme based on reinforcement learning (RL) and queuing theory (QT) in this paper.

This research work has the following theoretical and practical contributions.

(i) A novel dynamic resources provisioning scheme based on QT and RL is proposed.

(ii) Two concepts are introduced to evaluate the performance of various resources provisioning scheme.

(iii) To demonstrate our method, we apply our method to the simulation and real resources provisioning for cloud computing platform.

(iv) The experiment results demonstrate that our developed method can make accurate provisions at numerous arrival rates of users' job and avoid SLA conflicts.

The remainder of this paper is organized as follows. Section 2 reviews the related work of resources provisioning in cloud computing environment. Section 3 explores the construction of cloud computing platform and detailed analysis on the implementation of user job in proposed system model. Based on the cloud models, we introduce two new concepts: SSLA and UUTC, respectively. In Section 4, we design a resources allocation scheme based on reinforcement learning; then according to the shortage of basic Q learning, we propose an improved Q learning scheme to enhance the algorithm performance. The experiment results are presented in Section 5, and, finally, we reach the conclusions and skeleton of our future work in Section 6.

2. Related Work

Dynamic resource management in cloud computing environment refers to the process of dynamic optimization allocation, organization, coordination, and control of resources. It should not only support task scheduling in interorganizational or management domain, real-time resources monitoring, and job execution, but also maintain the self-management of the local sites, providing the corresponding Quality of Service (QoS) support. It is an advanced form of resource management and also the core component of resource management system in cloud computing environment, so as to shield the heterogeneity and complexity of the underlying resources, to manage the distributed massive resources in the cloud computing, to control the resources effectively, to

improve resource utilization, and to provide the reasonable distribution of resources operation for cloud computing, thus to balance the load.

Since the concept of cloud computing was proposed, resource scheduling, especially dynamic resources provisioning, has been one of the most important research components. Related works of resource provisioning are mainly from different perspectives to construct a cloud computing system model of queueing theory, aiming to attain universal results. However, being affected by factors such as the heterogeneity of the built platforms, the incompatible interface, and the disparity of the underlying physical resources, various research results demonstrate discrepancies, difficult to make a comparable analysis among each other.

In [2], dealing with the combined issues of power and performance management in cloud data centers, the authors proposed a dynamic resource management scheme by leveraging both of the techniques such as dynamic voltage/frequency scaling and server consolidation, thus to achieve energy efficiency and desired application-level performance. The novelty of the proposed scheme was its integration with timing analysis, queuing theory, integer programming, and control theory techniques. In [3], despite the varying event arrival rates, a queuing theory based approach was pursued to achieve specified response time target; by drawing the necessary computing resources from a cloud, a distinct query engine was modeled as an atomic unit to predict response times. Several similar units hosted on a single node were modeled as a multiple class M/G/1 queuing system and the response times were deemed to meet specified targets although being subject to varying event arrival rates over time. Correlation work was also extended to multimedia cloud and large web server clusters [4, 5]. In [4], concentrating on resource allocation problems in multimedia cloud, the authors employed optimization methods and queueing theory; theoretical analysis and computer simulation demonstrate the resource cost minimization problem and the response time minimization problem, respectively. In [6], the authors proposed a dynamic resource allocation scheme to resist distributed denial of service (DDoS) attacks against individual cloud customers. This paper was an early work that discussed mitigating DDoS attacks using resource allocation scheme for individual cloud customers. In [7, 8], the authors proposed an embedded Markov chain analytical model to estimate cloud performance. In [9], the authors focused on SLA-aware service deployment optimization problem, the designed E3-R which is a multiobjective genetic algorithm, to seek individuals and exhibit in cloud computing environments. In order to meet both requirements, E3-R employed two different fitness functions for different kinds of individuals. In [10], the authors focused on the bottleneck of network I/O and the aggregation on the packets delay and proposed a mechanism based on packet aggregation to achieve the best tradeoff between the throughput and packets delay. In [11], the authors investigated the elastic resource provisioning problem under the burstiness of incoming requests and energy consumption, employed the ON-OFF Markov chain and queueing theory to describe burstiness, and proposed a VM consolidation mechanism for each PM to solve the problem. In [12], the authors focused on cloud backup optimization problem, employing two decision parameters and finite-source queueing theory to maintain the regulated service quality of the cloud platform.

Distinguished from prior works, we establish a model of cloud computing system to strengthen the learning as an optimization tool for dynamic resources provisioning in data center based on the queueing theory in this paper. To the best of the authors' knowledge, not many related papers have appeared in the literature concerning dynamic resources provisioning.

3. System Model

3.1. Cloud Computing Platform Framework. In this section, we will introduce the framework of the cloud computing platform used in this study, as depicted in Figure 1. The details of system organization structure and the functions of various parts are described as follows.

Users Interface (UI). The main function of users interface (UI) is to receive the user's requests, allocating them to the corresponding VM cluster on the basis of provisioning algorithm and then receiving the execution results and returning them to the user.

Users Job Queue (UJQ). Cloud computing platform includes two user-job queues. After they are submitted via the user interface to the cloud computing platform, the jobs enter queue 1 in turn and wait to be scheduled. When they are executed, they enter the user queue 2 in turn, waiting to be sent back to the end user by the transmitter.

Users Job Scheduling (UJS). By employing the job scheduling policy, the user's requests in queue 1 are scheduled to the corresponding VM clusters.

Virtual Machine Cluster (VMC). A plurality of VMs to perform the same operation type forms a VMC. Each VM of a VMC is especially designed for a certain type of operation to enable the highest operating efficiency. Each VMC is equipped with a Virtual Machine Cluster Agent (VMCA), responsible for the VM instance generation, management, and cancellation. For example, when a user job arrival rate increases, the VMs within the current VMC cannot meet QoS or SLA, so VMCA needs to increase the number of the VMs within the VMC to improve its throughput; otherwise, some of the VMs in the VMC need to be cancelled so as to reduce energy consumption.

VM. Responsible for specific job execution, VM removes a user job from the queue for execution, transmits the execution results to the platform interface, and extracts the next user request. Each VM is equipped with a Performance Monitor Agent (PMA) and a Resource Management Agent (RMA). PMA is responsible for the performance indicators to monitor the entire VM, including response time, throughout, and resource utility. RMA is responsible for the VM resource management, mainly including the dynamic scheduling of CPU, memory, bandwidth, and data center resources.

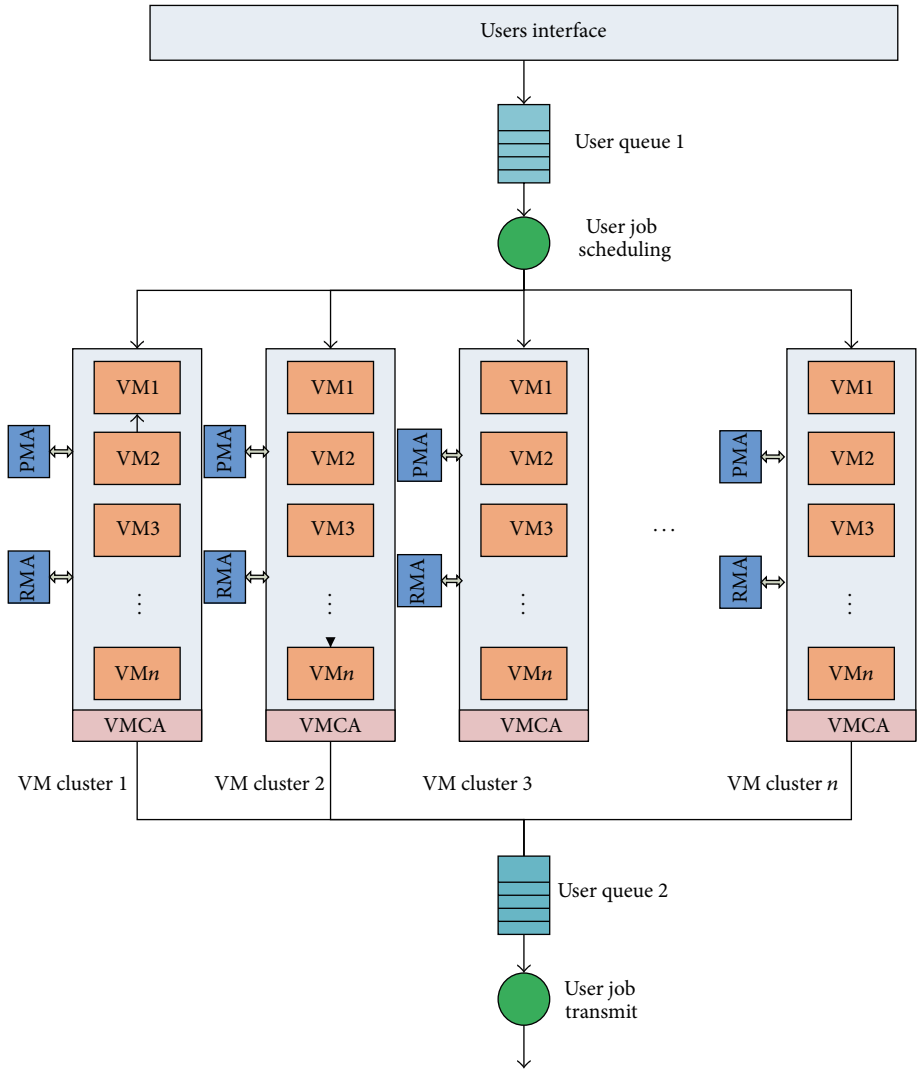


Figure 1: Architecture of cloud computing framework.

Users Job Transmit (UJT). The execution results of queue 2 are transmitted to the corresponding user in light of the transmission strategy.

3.2. Job Response Time (JRT). According to the system model shown in Figure 1 and the job execution phase in cloud computing environment, Job Response Time (JRT) hinges on Job Queueing Time (JQT), Job Execution Time (JET), and Job Transfer Time (JTT). In other words, JRT is made up of JQT, JET, and JTT.

In light of the classic queueing theory, given the fact that the job arrival rate of the computing platform is λ, the arrival rate of the jth VM in the cluster i is λ_{ij}, and its service rate is μ_{ij}. Thus, the allocated average queueing time JQT can be described as [13]

$$\mathrm{JQT} = \frac{\rho_{ij}}{\mu_{ij}\left(1-\rho_{ij}\right)}, \tag{1}$$

where $\rho_{ij} = \lambda_{ij}/\mu_{ij}$ and the Probability Density Function (PDF) of JQT is

$$f_{\mathrm{JQT}}(t) = \left(1-\rho_{ij}\right)\delta(t) + \mu\rho_{ij}\left(1-\rho_{ij}\right)e^{-\mu_{ij}(1-\rho_{ij})t} \quad (t \geq 0). \tag{2}$$

Similarly, the response time of JET and JTT [14, 15] is

$$\mathrm{JET} = \frac{1}{\mu_{ij}}, \qquad \mathrm{JTT} = \frac{D_{ij}/B_{ij}}{1-\lambda_{ij}D_{ij}/B_{ij}}, \tag{3}$$

where D_{ij} is the result size of user job and B_{ij} is the provisioned bandwidth resources of the user job. And the PDF of JET and JTT is

$$f_{\mathrm{JET}}(t) = \mu_{ij} e^{-\mu_{ij} t} \quad (t \geq 0), \tag{4}$$

$$f_{\mathrm{JTT}}(t) = \frac{B_{ij}}{D_{ij}} e^{-(B_{ij}/D_{ij})t} \quad (t \geq 0), \tag{5}$$

respectively.

Thus the total response time in the VMC can be given as

$$\begin{aligned} T_{\mathrm{tot}} &= \mathrm{JQT} + \mathrm{JET} + \mathrm{JTT} \\ &= \frac{\rho_{ij}}{\mu_{ij}\left(1 - \rho_{ij}\right)} + \frac{1}{\mu_{ij}} + \frac{D_{ij}/B_{ij}}{1 - \lambda_{ij} D_{ij}/B_{ij}}. \end{aligned} \tag{6}$$

3.3. Segment SLA. The response time, which is the performance indicator of the cloud computing platform, is constrained by QoS or SLA. In this study, we divide SLA into the varying phases at which the user job is to be executed, so as to provision the resources accurately in the cloud computing environment. When the job is run, the resources can be provisioned according to the different stages by the resource provisioning scheme, thus to enable each phase to be constrained by SLA, as indicated in the following representation:

$$\begin{aligned} \mathrm{JQT} &\leq \mathrm{SLA}_{\mathrm{JQT}}, \\ \mathrm{JET} &\leq \mathrm{SLA}_{\mathrm{JET}}, \\ \mathrm{JTT} &\leq \mathrm{SLA}_{\mathrm{JTT}}. \end{aligned} \tag{7}$$

As long as the job at each phase in the execution meets SLA constraints, the total response time is enabled to satisfy the global SLA constraints. Moreover, the introduction of segment SLA can improve the QoS of the cloud computing platform effectively. For instance, when a JQT violates SLA because of a resource shortage, I/O deadlock, or conflicts, a higher priority is given to the job execution in the upcoming phases, guaranteeing the resources for the job and reducing the corresponding time of JET and JTT, respectively, thus to ensure the overall SLA of the user operation to satisfy the QoS constraints. An example of medical image analysis application in cloud computing environment is shown in Figure 2.

3.4. Utility Unit Time Cost. Currently the commercial cloud computing platforms are mostly paid by being rented per hour. Take the well-known Amazon EC2 cloud computing platform for example, in which the price for standard on-demand instances is illustrated in Table 1.

As for any user job, the Utility Unit Time Cost (UUTC) can be defined as follows:

$$\mathrm{UUTC} = \frac{\text{Total cost}}{T_{\mathrm{tot}}}. \tag{8}$$

Physically, UUTC is the ratio of the operation cost and the actual execution time. It optimizes the constraints in terms of resource utilization rate and improves the optimized function.

Table 1: Amazon EC2 pricing for standard on-demand instances.

Instance type	Linux (per hour)	Windows (per hour)
Small (default)	\$0.060	\$0.115
Medium	\$0.120	\$0.230
Large	\$0.240	\$0.460
Extra large	\$0.480	\$0.920

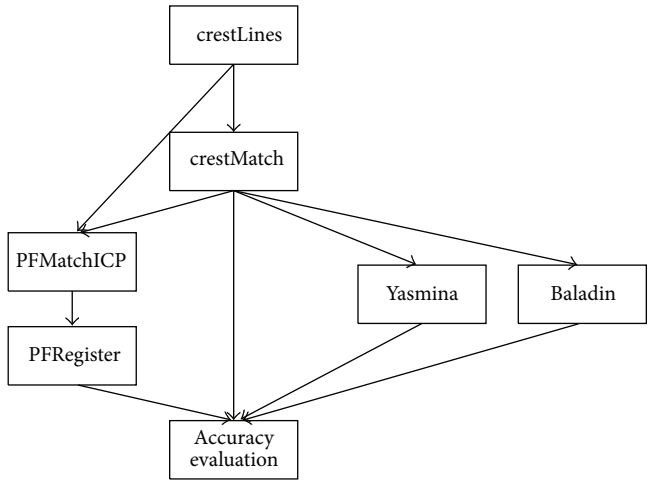


Figure 2: Medical image analysis application.

As for a user's job, the issue on resource provisioning optimization in cloud computing platform can be denoted as

$$\begin{aligned} \underset{\{\forall \text{user job}\}}{\text{Maximum}} \quad & \mathrm{UUTC} \\ \text{subject to} \quad & \mathrm{JQT} \leq \mathrm{SLA}_{\mathrm{JQT}} \\ & \mathrm{JET} \leq \mathrm{SLA}_{\mathrm{JET}} \\ & \mathrm{JTT} \leq \mathrm{SLA}_{\mathrm{JTT}}. \end{aligned} \tag{9}$$

As shown in (9), RAM makes a decision through attaining the performance index at every observant moment on the basis of PAM. Therefore, it is a sequential decision-making problem. Aiming to solve this problem, we propose a scheme in this study by employing reinforcement learning, which is described in detail in Section 4.

4. Resources Provisioning Mechanism

As stated in Section 3, the resource provisioning issue can be viewed as a sequential decision-making problem in the system model; therefore, it can be represented by MDP. Meanwhile, with little difference in the definition for the concepts like state space, action set, and reward function in the various RL-based schemes of resource provisioning in the cloud computing environment, we also define the relevant concepts as follows in this study.

4.1. Concepts Declared

4.1.1. State Space. A physical machine can virtualize a number of VMs, while a VM can only belong to one physical machine. The virtual ones are logically independent within the same physical machine, while they compete with each other in resource provisioning. Resources including VCPU, memory, and bandwidth, in each VM, are regarded as the state space in this study; accordingly, the state space for each VM is expressed in the form of a vector as VCPU, memory, and bandwidth. The value of every element is not beyond the upper bound of physical machine. Supposing a physical machine has four CPUs, 8 G memory, and 100 M bandwidth, an example of state space in VM is $(1, 2, 2)$, which means the VM has 1 VCPU, 2 G memory, and 2 M bandwidth.

4.1.2. Action Space. As for the ith VM resource, a possible action space includes the increase, constancy, or decrease in resource, which can be identified by $1, 0, -1$, respectively, to indicate the corresponding action. Meanwhile, the increased or decreased resource of VCPU, memory, and bandwidth is set as a VCPU, 512 M memory, and 0.5 M bandwidth at each decision-making moment. Then for the ith VM, assuming that its state space is $(1, 2, 2)$, the action at the decision moment can be expressed as $(0, 1, -1)$, which means the number of VCPU remains unchanged, memory increases to 512 M, and the bandwidth reduces to 0.5 M. After the action is implemented, VM's state space is represented as a vector $(1, 2.5, 1.5)$.

4.1.3. Immediate Reward. The immediate reward is used to reflect the correct running state and the efficiency of job scheduling. The three situations are considered by designing a reward function. (1) If the UUTC of current users job is bigger than the mean UUTC and satisfies SAL or QoS constraint, the reward is 1; (2) if the response time of users job violates SLA or QoS constraint, the reward is –1; (3) otherwise, the reward is 0.

4.2. Basic Reinforcement Learning Resources Allocation Scheme. As is depicted in the MDP, we employ the Q learning algorithm, a popular reinforcement learning algorithm, to solve the sequential decision problem described in (9). The pseudocode of the basic Q value learning algorithm is illustrated in Algorithm 1.

In order to evaluate the performance of the proposed resources provisioning scheme, we compare it to the utilization ratio provisioning scheme [16] used in the Amazon cloud computing platform. Due to the numerous experimental results, the ones listed here are only the results relevant to the VCPU resource provisioning, as shown Figure 3. In order to make our simulations more convincing, we use the practical parameters and pricing rates of Windows Azure. Windows Azure is a cloud platform developed by Microsoft, which provides on-demand computation and bandwidth resources for services through Microsoft data centers.

We assumed the increasing arrival rate of different jobs and the number of user jobs to be completed at each arrival rate to be the same in the experiments. As seen in Figure 3(a), based on the different demands for specific resources of different jobs, when the job arrival rate increases, the utilization strategy can make a real-time provision with VCPU resources in virtual machines efficiently and meanwhile avoid the SLA conflict as shown in Figure 3(b). However, under the same experimental conditions, basic Q learning scheme adjusts VCPU resources provisioning frequently, which may result from a wrong performance (e.g., when it is the time to increase VCPU resources, they are not raised but reduced instead) due to the exploration-exploitation mechanism, eventually leading to the frequent resource provisioning and SLA conflicts.

(1) **Initialize** Q value table
(2) **Initialize** state s_t
(3) error = 0
(4) **repeat**
(5) **for** each state s **do**
(6) a_t = get_action(s_t) using ε-greedy policy
(7) **for** (step = 1; step < **LIMIT**; step++) **do**
(8) **take action** a_t observe **r** and S_{t+1}
(9) $Q_t = Q_t + \alpha * (r + \gamma * Q_{t+1} - Q_t)$
(10) error = MAX(error | $Q_t - Q_{\text{previous-}t}$)
(11) $s_t = s_{t+1}$, a_{t+1} = get_action(s_t), $a_t = a_{t+1}$
(12) **end for**
(13) **end for**
(14) **Until** error < θ

Algorithm 1: Q value learning algorithm.

From the comparison results in Figure 3, we can draw some conclusions of basic reinforcement learning used in cloud computing resources provisioning.

(i) Slow convergence.

(ii) Ineffective adaptability to the changeable arrival rates; thus the policy needs updating, and sometimes not getting the converged solution.

(iii) The suboptimal solution is often acquired instead of the optimal solution, even at a fixed job arrival rate.

The above disadvantages of reinforcement learning, especially the slow convergence rate and ineffective adaptive ability, severely limit its practical application in the cloud resource provisioning.

4.3. Improved Reinforcement Learning Resources Allocation Scheme. We design an improved scheme to conquer each weakness of basic reinforcement learning scheme, which is focused on the following aspects.

4.3.1. Offline Learning. With the simulations to the real data sets, the offline training based on the basic Q learning algorithm is employed to acquire the varied job arrival rates, the numbers of the VMs, and the Q value table, the approximate function relation between resource provisioning. During offline learning process, multiple instances can run parallelly in order to learn by dividing the state space, so

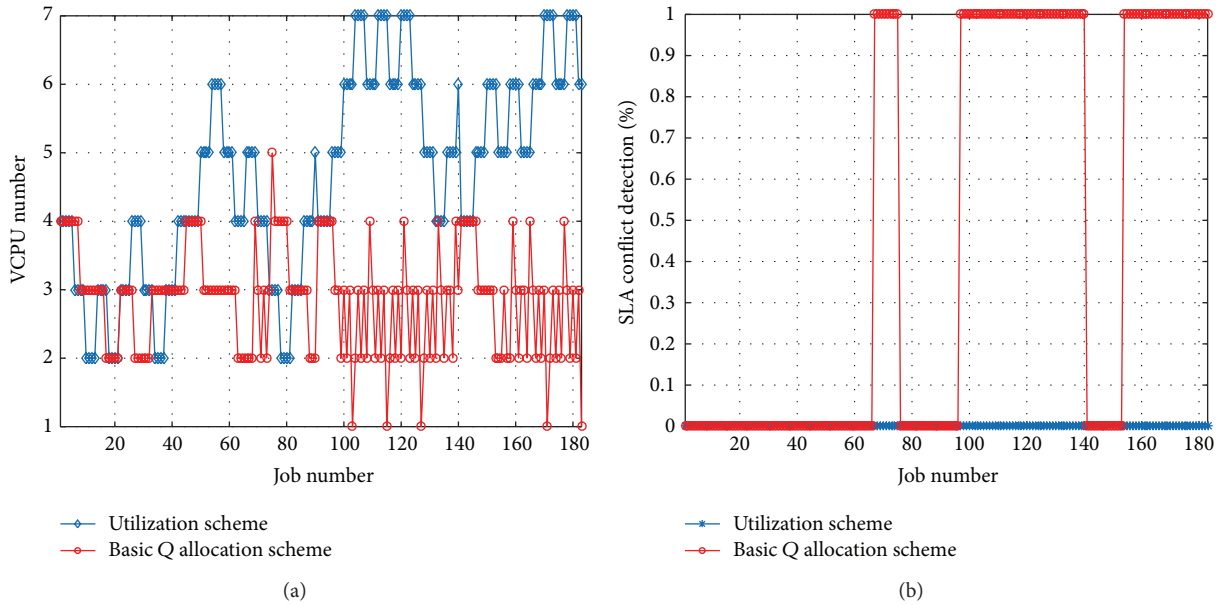


Figure 3: Comparison results of VCPU resource provisioning and SLA conflict detection between the basic Q learning scheme and the utilization scheme. (a) Comparison of VCPU number between the basic Q learning scheme and the utilization scheme under various job numbers. (b) SLA conflict detection between the basic Q learning scheme and the utilization scheme under various job numbers.

(1) **Divide** State Space
(2) **Repeat**
(3) **for** each state space partition **do**
(4) **set** upper and low bound of CPU, memory and bandwidth
(5) **Obtain** running state of cloud computing platform
(6) **Obtain** performance index
(7) **for** each resources **do**
(8) using Algorithm 1
(9) **end for**
(10) **end for**
(11) **Update** Q value table

Algorithm 2: Offline reinforcement learning algorithm.

(1) **Obtain** running state of cloud computing platform
(2) **Look up** Q value table, configure VM resources
(3) **Obtain** performance index
(4) **use** belief library
(5) **set** upper and low bound of VCPU, memory and bandwidth
(6) action space compact
(7) **for** each resources **do**
(8) using Algorithm 1
(9) **end for**
(10) **Update** Q value table

Algorithm 3: Online reinforcement learning algorithm.

that the acquired relation can be approximately formulated with the regression function. The pseudocode of the Q value offline learning algorithm is illustrated in Algorithm 2.

In spite of the fact that the Q value table resulting from offline learning is rather huge, data index can be used to accelerate the search speed, thus improving search efficiency.

4.3.2. Belief Libraries and Simple Action Space. Set up a VCPU belief library, whose rules are similar to the ways by which the belief library was built in [17]. Based on the established belief library, the action space can be simplified correspondingly. For example, when provisioning resources, if the VCPU utilization approaches the lower bound, the action increase to raise up VCPU resources in action space should be removed; otherwise the decrease to reduce the VCPU resources should be removed. The establishment of belief library and the simplification of action space avoid the blindness of Q learning action selection effectively and thus improve the convergence speed.

4.3.3. Online Learning. Offline learning environment can only simulate part of the real operating environment. The acquired function is a viable strategy for resource provisioning only when a job arrival rate meets the SLA constraint; even so, it may not be a suboptimal strategy. However, this initial strategy sets up an upper boundary for resource provisioning, and under the guidance of it we can learn online based on this initial strategy to further improve the resource utilization.

The acquired real-time resource utilization rate by using PMA guides RMA learning. The pseudocode of the Q-value online learning algorithm is illustrated in Algorithm 3.

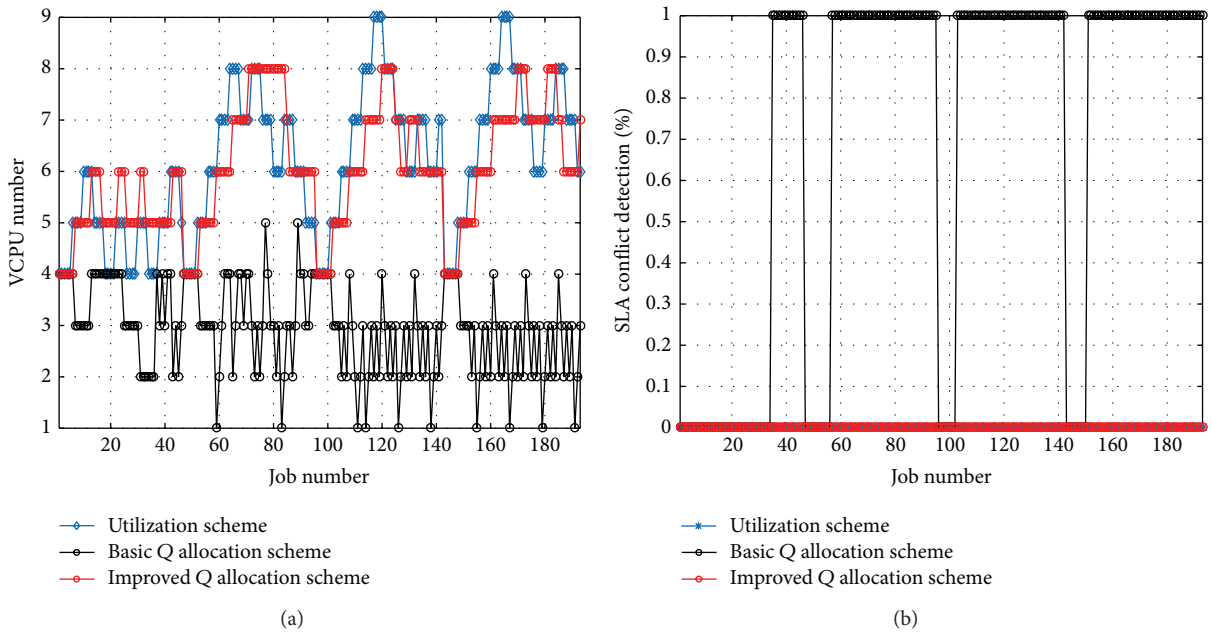


Figure 4: Comparison results of VCPU resource provisioning and SLA conflict detection between the improved Q learning scheme, the basic Q learning scheme, and the utilization scheme. (a) Comparison of VCPU number between the improved Q learning scheme, the basic Q learning scheme, and the utilization scheme under various job numbers. (b) SLA conflict detection between the improved Q learning scheme, the basic Q learning scheme, and the utilization scheme under various job numbers.

5. Experience Results

To evaluate the efficiency of our approach, implementations have been performed on the simulation and real cloud computing environment, respectively.

5.1. Simulation Experiment Results. Using MATLAB R2012a by MathWorks, Inc., we have developed a discrete event simulator of the cloud server form to validate the efficiency of resource provisioning solution and have compared the performance information among the alternative schemes in our simulations.

We evaluated the performance of the improved Q learning strategy in Figure 4 and compared it with the utilization rate scheme and the basic Q learning scheme under the same experimental conditions in Section 4.2. From the experimental results in Figure 4, on one hand, we can see that basic Q learning scheme is still likely to make wrong decisions and thus resulted in the SLA conflicts and frequent VCPU resource provisions. On the other hand, the improved Q learning scheme can provision the VCPU resources in real time based on the changes in arrival rates; more importantly, apart from avoiding SLA conflicts, the number of VCPU resources used by it appears less than that used by utilization rate scheme through most of the time. In other words, avoiding the SLA conflict, the improved Q learning method improves the utilization rate of resources.

Next, we compare the improved Q learning scheme with the prevalent resource provisioning schemes: (1) the proposed resources provisioning scheme, denoted by the improved Q learning scheme, in which the cloud computing resources were optimally scheduled to the VMs by improved Q learning algorithm, (2) the utilization resources provisioning scheme, denoted by utilization scheme, in which the cloud computing resources were optimally scheduled to the VMs by resources utilization, (3) the genetic algorithm resources provisioning scheme, denoted by GA scheme, in which the cloud computing resources were optimally scheduled to the VMs by genetic algorithm, (4) the nonlinear programming resources provisioning scheme, denoted by nonlinear programming scheme, in which the cloud computing resources were optimally scheduled to the VMs by nonlinear programming.

As seen from the experimental results in Figure 5(a), the number of VCPU resources in the GA scheme and the nonlinear programming scheme may cause frequent changes due to the objective function optimization. Nevertheless, the improved Q learning scheme and the utilization scheme demonstrate the same performances as those in Figure 4(a), respectively. With similar pricing settings to Table 1, Figure 5(b) shows that the total cost of the improved learning Q strategy designed in this paper is lower than that of the compared schemes.

We ran the simulation program 1000 times at various arrival rates, and the average results of various performance indicators are shown in Table 2. As shown in the results, the UUTC at different arrival rates is better than that of the contrast algorithm. In light of the definition of UUTC, the numerator aims to maximize the current operation costs, while the denominator aims to minimize the current execution time. The ratio between the two indicates the utilization rate of cloud computing resources per unit time. Accordingly, the greater the value is, the higher the corresponding utilization rate of the cloud computing resource is.

Table 2: Comparison results of various resources allocation schemes.

	Arrival rate (jobs per min.)	Total running time (min.)	Total cost (dollars)	Mean VCPU numbers (numbers per arrival rate)	UUTC (dollars per min.)
RPSUT	30	364.61	7270	7.36	2.70
RPSGA		381.68	3930	7.36	1.39
RPSNP		368.36	4353	7.36	1.60
Proposed scheme		368.36	5204	4.76	**2.96**
RPSUT	40	247.20	7433	8.20	3.66
RPSGA		273.12	5788	8.20	2.58
RPSNP		278.21	5349	8.20	2.34
Proposed scheme		250.30	3727	3.87	**3.84**
RPSUT	50	216.14	8688	8.18	4.91
RPSGA		244.06	8100	8.18	4.05
RPSNP		232.41	6376	8.18	3.35
Proposed scheme		218.63	5448	4.45	**5.59**
RPSUT	60	148.39	9540	8.78	7.32
RPSGA		172.75	8718	8.78	5.74
RPSNP		176.30	7569	8.78	4.88
Proposed scheme		149.85	6578	4.43	**9.90**
RPSUT	70	147.48	9611	9.96	6.54
RPSGA		175.78	10025	9.96	5.72
RPSNP		179.37	9878	9.96	5.529
Proposed scheme		147.87	7820	7.69	**6.87**

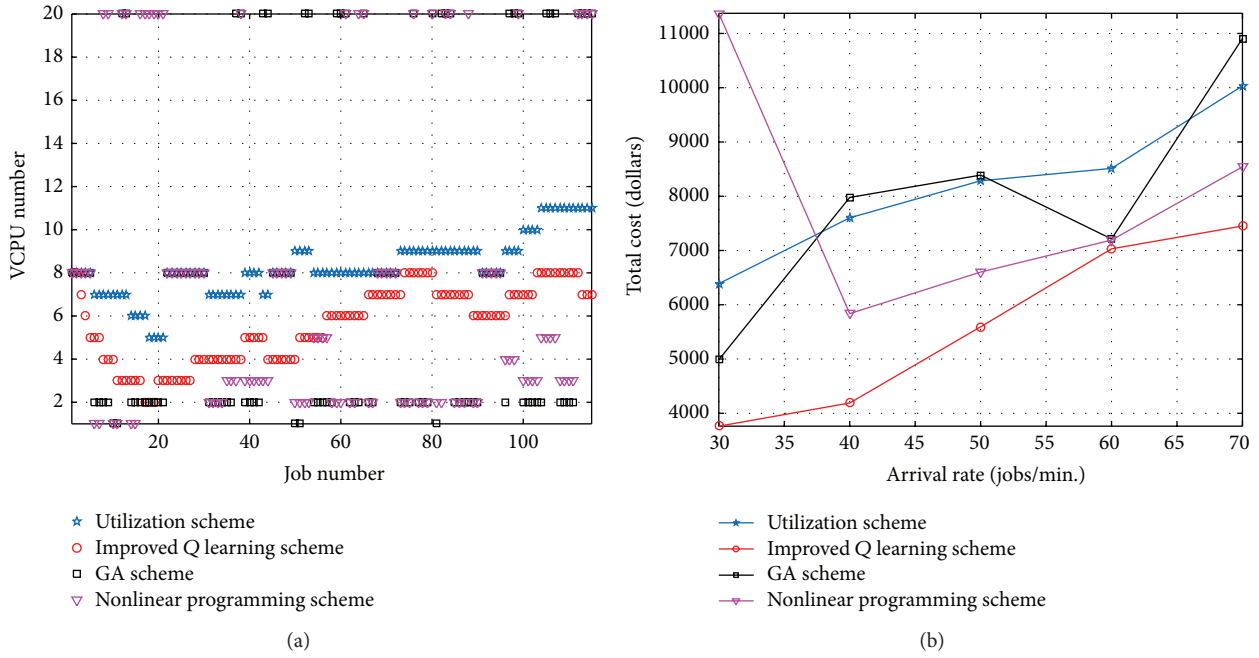


Figure 5: Comparison results of VCPU resource provisioning and total cost between the improved Q learning scheme, GA provisioning scheme, the utilization scheme, and nonlinear programming provisioning scheme. (a) Comparison of VCPU number between the improved Q learning scheme, GA provisioning scheme, the utilization scheme, and nonlinear programming provisioning scheme under various job numbers. (b) Comparison of total cost between the improved Q learning scheme, GA provisioning scheme, the utilization scheme, and nonlinear programming provisioning scheme under various job arrival rates.

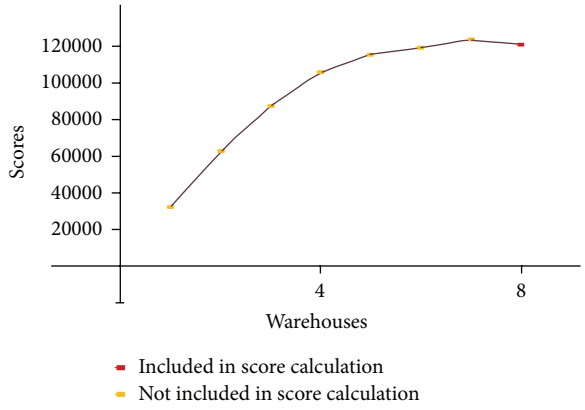


Figure 6: SPECjbb2005 throughput in default settings.

The utilization scheme in the figure executes the jobs with the real-time resource provisioning based on the utilization. It is demand-sensitive to the resources and requires frequent application and release of the resources. Based on the job arrival rates, the improved Q provisioning scheme can not only configure the system resources in accordance with Q value table optimization scheme, but also learn the optimal provisioning further on the real-time resource utilization. The improved Q allocation scheme can configure the cloud computing resources based on the arrival rates of the user's job and learn the optimal allocation further based on the optimization scheme. This scheme facilitates the adaptability to the user's job arrival rates, while the basic Q learning scheme needs some exploration before it gets steady, or even worse, it cannot acquire the stable distribution. From the experiment results in Figure 5, a similar conclusion is drawn, which further proves the effectiveness of this provisioning scheme.

5.2. Real Cloud Computing Platform Experiments Results. The machines used in the experiments consist of virtual servers, client, and compute machines. The physical machines for virtual hosting are Lenovo ThinkServer RD630 with 8 CPU and 16 GB memory.

Xen was used as our virtualization environment and the SPECjbb2005 was selected as the workloads running within the VMs. SPECjbb2005, a Java program, is SPEC's benchmark. By simulating a three-tier client/server system with stress on the middle tier, the benchmark measures the performance of server side Java. It implements the Java virtual machine (JVM), just-in-time (JIT) compiler, garbage collection, threads, and part of the operating system. Implemented in a more object-oriented manner, SPECjbb2005 presents new features such as XML processing and BigDecimal computations and furnishes a new enhanced workload to mirror how real application is designed, thus making the benchmark a more realistic reflection of contemporary application [18].

The SPECjbb2005 throughput under various pressures in the default setting can be seen in Figure 6. The default setting in SPECjbb2005 refers to the whole resources in the platform used by the system under any pressure, such as CPU and memory. In the experiments processing, warehouses were dynamically increasing from 1 to 8 every fixed time interval. In our experiments, the time interval was set as 20 minutes. Another fact discovered in the experiments at the same time was that the benchmark was insensitive to memory. Hence, only the results of CPU resources were listed in the next part.

Table 3: Comparison results of average throughput under various warehouses (Bops).

Warehouse	Max	Utilization scheme	Proposed scheme
1	32103.43	32847.88	33767.96
2	63252.47	59510.64	59951.13
3	87267.56	82380.99	83919.59
4	105719.64	104081.37	99212.12
5	115648.76	117200.12	116671.41
6	119057.39	121530.59	122071.97
7	123759.19	126187.14	124289.81
8	120844.70	122457.18	122758.30

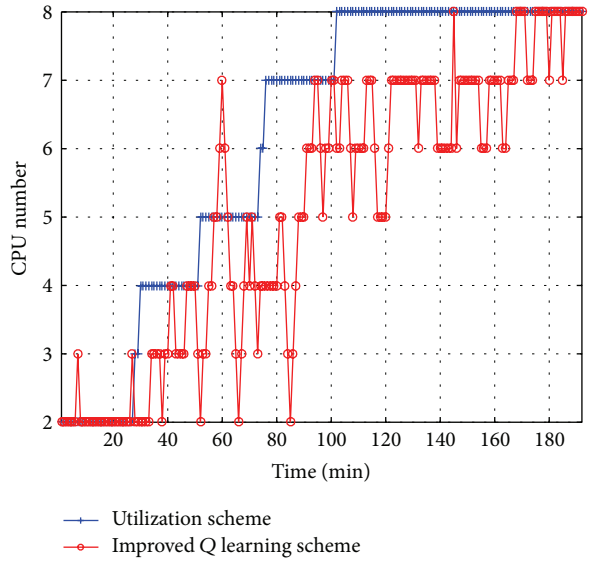


Figure 7: Comparison of CPU number between the improved Q learning scheme and the utilization scheme under various warehouses.

We ran the benchmark 10 times at various warehouses, and the comparison results of average throughput under various warehouses were shown in Table 3. From the results we can see the throughput of the utilization scheme and the proposed scheme was close to the reference Max under various warehouses. It proves these two resource provisioning schemes can provision CPU resources self-adaptively and achieve the maximum throughput according to different warehouses.

The real cloud computing environment experiment results as shown in Figure 7 also demonstrated that the proposed improved Q provisioning scheme outperforms the utilization scheme in that it achieved resources utilization rate under various warehouses and CPU resources constraint.

6. Conclusions and Future Work

In this study, we provide an insightful view about the resource provisioning optimization problem in cloud computing platform; then we propose a novel resources provisioning scheme based on the reinforcement learning and queueing theory. With the introduction of the concepts of SSLA and UUTC, we view the resource provisioning issue in cloud computing as a sequential decision problem, and then we design a novel optimization object function and employ reinforcement learning to solve it. Experiment results not only demonstrate the effectiveness of the proposed scheme, but also prove to outperform the prevalent resource provisioning methods in terms of SLA collision avoidance and user costs. Also, some conclusions can be drawn as below for using Q learning algorithm.

(i) The Q learning algorithm outperforms the comparative method in three aspects: the number of CPUs in use, SLA conflicts, and costs.

(ii) Belief library is employed to simplify the action space, thus to enhance the performance of the Q learning algorithm.

(iii) State space is the number of VCPU/CPU in experiment, with a smaller space, faster convergence, and stronger adaptability.

(iv) Q learning provisions in two cases, either when the SLA violation occurs for three consecutive times, or when the utilization rate is too high or too low.

In the future, we plan to extend our schemes for the dynamic resource provisioning to get the minimal response time, thus providing highly satisfactory services and avoiding SLA violations. Looking into the cloud entities and considering the details of cloud computing platform, such as VMs failures, VMs migration, costs of communication, and burst arrivals of requests, VMs cluster for different kinds of requests will be other dimensions of extension.

Abbreviations

UI: Users interface
UJQ: Users job queue
UJS: Users Job Scheduling
UJT: Users Job Transmit
VM: Virtual Machine
VMC: Virtual Machine Cluster
VMCA: Virtual Machine Cluster Agent
PMA: Performance Monitor Agent
RMA: Resource Management Agent
JRT: Job Response Time
JQT: Job Queueing Time
JET: Job Execution Time
JTT: Job Transfer Time
QoS: Quality of Service
SLA: Service Level Agreement
SSLA: Segmentation SLA
UUTC: Utility Unit Time Cost
MDP: Markov decision process
PDF: Probability Density Function
ML: Machine learning
RL: Reinforcement learning
λ: Job arrival rate of cloud computing platform
λ_{ij}: Job arrival rate of jth VM in ith VMC
μ_{ij}: Job server rate of jth VM in ith VMC
D_{ij}: The executed result size of user job
B_{ij}: The allocation bandwidth resources of the user job.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The work presented in this paper was supported by National Natural Science Foundation of China (nos. 61272382 and 61402183).

References

[1] H.-S. Wu, C.-J. Wang, and J.-Y. Xie, "TeraScaler ELB-an algorithm of prediction-based elastic load balancing resource management in cloud computing," in *Proceedings of the 27th International Conference on Advanced Information Networking and Applications Workshops (WAINA '13)*, pp. 649–654, IEEE, Barcelona, Spain, March 2013.

[2] Y. Gao, H. Guan, Z. W. Qi, T. Song, F. Huan, and L. Liu, "Service level agreement based energy-efficient resource management in cloud data centers," *Computers and Electrical Engineering*, vol. 40, pp. 1621–1633, 2013.

[3] V. Suresh, P. Ezhilchelvan, and P. Watson, "Scalable and responsive event processing in the cloud," *Philosophical Transactions of the Royal Society A*, vol. 371, no. 1983, Article ID 20120095, 2013.

[4] X. Nan, Y. He, and L. Guan, "Queueing model based resource optimization for multimedia cloud," *Journal of Visual Communication and Image Representation*, vol. 25, no. 5, pp. 928–942, 2014.

[5] H. Nguyen, T. N. Minh, and N. Thoai, "Tool-driven strategies for resource provisioning of single-tier web applications in clouds," in *Proceedings of the 5th International Conference on Ubiquitous and Future Networks (ICUFN '13)*, pp. 795–799, July 2013.

[6] S. Yu, Y. Tian, S. Guo, and D. O. Wu, "Can we beat DDoS attacks in clouds?" *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2245–2254, 2014.

[7] K. Salah and R. Boutaba, "Estimating service response time for elastic cloud applications," in *Proceedings of the 1st IEEE International Conference on Cloud Networking (Cloud Net '12)*, pp. 12–16, Jussieu, Paris, November 2012.

[8] H. Khazaei, J. Misic, and V. B. Misic, "Performance analysis of cloud computing centers using M/G/m/m+r queuing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 5, pp. 936–943, 2012.

[9] H. Wada, J. Suzuki, Y. Yamano, and K. Oba, "Evolutionary deployment optimization for service-oriented clouds," *Software: Practice and Experience*, vol. 41, no. 5, pp. 469–493, 2011.

[10] M. Bourguiba, K. Haddadou, I. E. Korbi, and G. Pujolle, "Improving network I/O virtualization for cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 3, pp. 673–681, 2014.

[11] Z. Luo and Z. Qian, "Burstiness-aware server consolidation via queuing theory approach in a computing cloud," in *Proceedings of the 27th IEEE International Symposium on Parallel & Distributed Processing (IPDPS '13)*, pp. 332–341, IEEE, Cambridge, Mass, USA, May 2013.

[12] F.-C. Jiang, C.-T. Yang, C.-H. Hsu, and Y.-J. Chiang, "Optimization technique on logistic economy for cloud computing using finite-source queuing systems," in *Proceedings of the 4th IEEE International Conference on Cloud Computing Technology and Science (CloudCom '12)*, pp. 827–832, December 2012.

[13] G. Grimmett and D. Stirzaker, *Probability and Random Processes*, Oxford University Press, 3rd edition, 2010.

[14] Y. Wu, C. Wu, B. Li, X. Qiu, and F. C. M. Lau, "CloudMedia: when cloud on demand meets video on demand," in *Proceedings of the 31st International Conference on Distributed Computing Systems (ICDCS '11)*, pp. 268–277, IEEE, Minneapolis, Minn, USA, July 2011.

[15] J. Zheng and E. Regentova, "Qos-based dynamic channel allocation for GSM/GPRS networks," in *Network and Parallel Computing*, vol. 3779 of *Lecture Notes in Computer Science*, pp. 285–294, Springer, Berlin, Germany, 2005.

[16] A. Wolke and G. Meixner, "Twospot: a cloud platform for scaling out web applications dynamically," in *Proceeding of 3rd European Conference on Service Wave*, pp. 13–24, Ghent, Belgium, 2010.

[17] X. Bu, J. Rao, and C.-Z. Xu, "A reinforcement learning approach to online web systems auto-configuration," in *Proceedings of the 29th IEEE International Conference on Distributed Computing Systems Workshops (ICDCS '09)*, pp. 2–11, Montreal, Canada, June 2009.

[18] http://www.spec.org/jbb2005/.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 594264, 13 pages
http://dx.doi.org/10.1155/2015/594264

Research Article

Modeling and Analysis of Queueing-Based Vary-On/Vary-Off Schemes for Server Clusters

Cheng-Jen Tang[1] and Miau-Ru Dai[2]

[1]*Department of Electrical Engineering, Tatung University, Taipei 10452, Taiwan*
[2]*Delta Network Inc., Taipei 11491, Taiwan*

Correspondence should be addressed to Cheng-Jen Tang; ctang@ttu.edu.tw

Received 7 January 2015; Accepted 30 May 2015

Academic Editor: Stefano de Miranda

A cloud system usually consists of a lot of server clusters handling various applications. To satisfy the increasing demands, especially for the front-end web applications, the computing capacity of a cloud system is often allocated for the peak demand. Such installation causes resource underutilization during the off-peak hours. Vary-On/Vary-Off (VOVO) schemes concentrate workloads on some servers instead of distributing them across all servers in a cluster to reduce idle energy waste. Recent VOVO schemes adopt queueing theory to model the arrival process and the service process for determining the number of powered-on servers. For the arrival process, Poisson process can be safely assumed in web services due to the large number of independent sources. On the other hand, the heavy-tailed distribution of service times is observed in real web systems. However, there are no exact solutions to determine the performance for *M/heavy-tailed/m* queues. Therefore, this paper presents two queueing-based sizing approximations for Poisson and non-Poisson governed service processes. The simulation results of the proposed approximations are analyzed and evaluated by comparing with the simulated system running at full capacity. This relative measurement indicates that the Pareto distributed service process may be adequately modeled by memoryless queues when VOVO schemes are adopted.

1. Introduction

The numbers of Internet requests are not uniformly distributed over time. There are a huge number of requests during the peak hours. Cloud service providers tend to install surplus server nodes to handle the bursty load. Clearly, these servers waste a lot of energy during the off-peak periods. Dynamically adjusting the number of active servers, that is, Vary-On/Vary-Off (VOVO) scheme, improves energy-efficiency of server clusters. However, overly shrinking the number of powered-on servers may lead to decreased service quality. Therefore, finding the *right* number of active servers to balance energy consumption and operation performance is a primary issue of an applicable VOVO scheme.

VOVO schemes can be dated back to earlier last decade [1, 2]. The basic idea of earlier VOVO schemes is to dynamically size a cluster according to CPU utilization or resource usage. This resource provisioning problem in a cluster can be analogous to the staff sizing problem in a telephone call center. In a call center, the customers are callers, servers are telephone agents, and tele-queues consist of callers that await service by an agent. The well-known *Erlang-C* model [3] has been widely applied to this problem. Many recent VOVO studies [4–9] adopt queueing analysis to manage resource usage of clusters.

Most available analytic solutions in queuing theory rely on independence assumptions and Poisson processes [10]. Internet traffic patterns are well known to possess extreme variability and bursty structure [11]. The heavy-tailed distributions of service times are observed in real web systems [12, 13]. This characteristic is characterized by self-similar process [14]. Pareto distribution is a popular model of self-similar processes [15]. However, queueing models with Pareto distributed service times are very difficult to analyze [16]. Although heavy-tailed service processes in web systems are widely documented, memoryless queues are still used for evaluating system performance in many studies [17–21]. On the other hand, studies [22–25] that adopt general/Pareto distributions need approximations for the analytically intractable distributions to obtain the performance measures.

The Poisson arrival process is particularly appropriate if the arrivals are from a large number of independent sources [10], such as users of web services. However, exploring the difference between modeling service times with Poisson process and non-Poisson process governed queues remains a challenging research topic, since many queueing models remain analytically intractable [26]. In order to understand the performance difference between modeling service times with Poisson process and non-Poisson process governed queues, a series of simulations are conducted in this study. Compared with the mathematical analysis and numerical methods, simulation is more time and memory consuming but it is sometimes the only way to get reasonably accurate results [27].

This paper presents the approximations of VOVO cluster sizing for systems modeled by $M/G/m$ and $M/M/m$. Randomly generated workload traces with Pareto and exponential distributed service times are simulated using the approximations. Two distinct types of real web access logs are simulated as well. A relative performance evaluation method is proposed and used for gauging the simulation results. Through the evaluation, the performance difference between modeling service times with Poisson process and non-Poisson process governed queues is found. The result suggests that $M/M/m$ based sizing approach may be adequate when a queueing-based VOVO scheme is adopted in a cluster.

This paper is organized as follows. Section 2 shows the approximation methods for cluster sizing. Section 3 details the simulation setup and the evaluation metric. Section 4 presents the simulation process and discusses the results. Section 5 concludes this paper.

2. Approximation for Queueing-Based Cluster Sizing

Investigations in a queueing theory applied system mainly aim at getting the performance measures, which are the probabilistic properties of the random variables, including number of customers in the system, number of waiting customers, utilization of the servers, response time of a request, waiting time of a customer, idle time of the server, and busy time of a server. These measures heavily depend on the assumptions concerning the distributions of interarrival times and service times as well as number of servers and service discipline. Queueing analysis can be naturally applied to the performance measures of server clusters. Server clusters have been widely adopted in many cloud data centers to resolve the increasing user needs [28]. Although heterogeneity is common in multifunctional cloud data centers, server closets or blade systems that form the basic computing units usually consist of homogeneous nodes. Therefore, this work focuses on single-queue homogeneous systems.

The symbols and definitions used in this paper to describe the performance measures of queueing systems are shown in Symbols and Definitions.

In classical queueing analysis, supposing that requests are handled by a single-queue m homogeneous server system with the First-Come First-Serve (FCFS) discipline, exponentially distributed service times, and Poisson process governed arrival intervals, the system can be modeled as $M/M/m$ system. ρ must be less than 1 ($\rho < 1$) for $M/M/m$ system being in a stable state. Many performance measures of a stable $M/M/m$ system have been thoroughly studied and are shown in (1) to (7). The calculations and proofs of these equations can be found in many textbooks, for example, [29, p. 412]:

$$p_j = \begin{cases} p_0 \dfrac{(m\rho)^j}{j!}, & \text{if } 0 < j < m \\ p_0 \dfrac{(m\rho)^j}{m! m^{j-m}}, & \text{if } j \geq m, \end{cases} \tag{1}$$

$$p_0 = \left(\sum_{j=0}^{m-1} \frac{(m\rho)^j}{j!} + \frac{(m\rho)^m}{m!(1-\rho)} \right)^{-1}, \tag{2}$$

$$E[N] = m\rho + \frac{\rho p_m}{(1-\rho)^2}, \tag{3}$$

$$E[M] = \frac{\lambda}{\mu}, \tag{4}$$

$$\begin{aligned} E[R] &= \frac{E[N]}{\lambda} = \frac{1}{\mu} + \frac{p_m}{m\mu(1-\rho)^2} \\ &= \frac{1}{\mu} + \frac{p_m}{m\mu(1-\lambda/m\mu)^2}, \end{aligned} \tag{5}$$

$$E[N_1] = \frac{\rho}{1-\rho}, \tag{6}$$

$$E[R_1] = \frac{1}{\mu(1-\rho)} = \frac{1}{\mu(1-\lambda/\mu)}. \tag{7}$$

2.1. Approximation for Sizing $M/M/m$ Modeled Clusters. In a homogeneous $M/M/m$ system, μ of each server is identical. From (5), $E[R]$ of $M/M/m$ system can be considered as a function of λ denoted by $f_m(\lambda)$:

$$f_m(\lambda) = \frac{1}{\mu} + \frac{p_m}{m\mu(1-\lambda/m\mu)^2}. \tag{8}$$

Let λ_m be an arrival rate of $M/M/m$ system maintaining a targeted response time r given $r > 1/\mu$. The curves of $f_m(\lambda)$ for $m = k-2$, $m = k-1$, $m = k$, $m = k+1$, and $m = k+2$ with a targeted response time r are shown in Figure 1.

λ_1 can be easily obtained from (5):

$$r = \frac{1}{\mu(1-\lambda_1/\mu)}, \tag{9}$$

$$\lambda_1 = \mu - \frac{1}{r}. \tag{10}$$

For $m > 1$, r, based on (7), can be represented as

$$r = \frac{1}{\mu} + \frac{p_m}{m\mu(1-\lambda_m/m\mu)^2}. \tag{11}$$

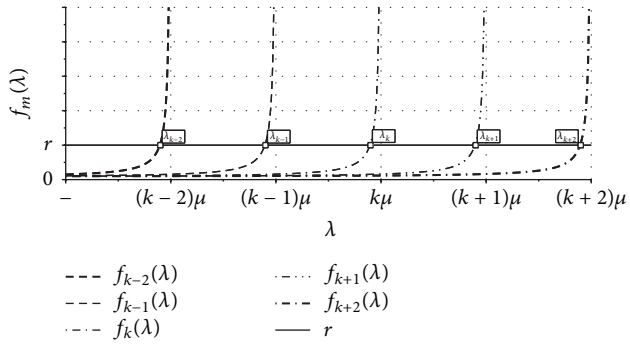


Figure 1: $f_m(\lambda)$ versus arrival rate (λ) with a targeted response time r.

From (1) and (2), p_m is

$$\begin{aligned} p_m &= p_0 \frac{\left(m\left(\lambda_m/m\mu\right)\right)^m}{m!} \\ &= \frac{\left(\lambda_m/\mu\right)^m/m!}{\left(\sum_{j=0}^{m-1}\left(\left(\lambda_m/\mu\right)^j/j!\right)+\left(\lambda_m/\mu\right)^m/m!\left(1-\lambda_m/m\mu\right)\right)}. \end{aligned} \tag{12}$$

Therefore, to get λ_m of μ for $m \geq 2$, the following equation has to be solved:

$$r = \frac{1}{\mu} + \frac{\left(\lambda_m/\mu\right)^m}{m!m\mu\left(\sum_{j=0}^{m-1}\left(\left(\lambda_m/\mu\right)^j/j!\right)+\left(\lambda_m/\mu\right)^m/m!\left(1-\lambda_m/m\mu\right)\right)\left(1-\lambda_m/m\mu\right)^2}. \tag{13}$$

λ_2 can also be easily obtained by solving (13) with $m = 2$:

$$\lambda_2 = 2\sqrt{\mu^2 - \frac{\mu}{r}}. \tag{14}$$

It is difficult to get a closed-form expression of λ_m in terms of r, μ, and m when $m > 2$. Therefore, an approximation is proposed for λ_m for $m > 2$. Assume that this approximation can be applicable for the systems with at most c servers. Every $f_m(\lambda)$, $c \geq m \geq 2$, is shifted with the offset value of $-(m-1)\mu$ and denoted as $g_m(\lambda)$:

$$g_m(\lambda) = f_m\left(\lambda + (m-1)\mu\right), \quad \text{for } c \geq m \geq 2. \tag{15}$$

Figure 2 shows the combination of the curves of $f_1(\lambda)$ and $g_m(\lambda)$, $c \geq m \geq 2$, with emphasis on the intersections between the targeted response time r and these curves.

By observing Figure 2, the distances between all consecutive $\lambda_{m-1} - (m-2)\mu$ and $\lambda_m - (m-1)\mu$ approximately form an exponential decay series $\{\delta_1, \delta_2, \ldots, \delta_c\}$. Let the series be approximated by an exponential decay function, let α be the initial quantity, and let β be the exponential decay constant. An element δ_m in the series can be expressed as

$$\delta_m = \alpha e^{-\beta(m-1)} = \lambda_{m-1} - \lambda_m + \mu. \tag{16}$$

Let the initial quantity $\alpha = \delta_1$; α can be obtained from (10):

$$\alpha = \mu - \lambda_1 = \mu - \left(\mu - \frac{1}{r}\right) = \frac{1}{r}. \tag{17}$$

From (17), (16), (10), and (14), δ_2 is

$$\delta_2 = \lambda_1 - \lambda_2 + \mu = 2\mu - \frac{1}{r} - 2\sqrt{\mu^2 - \frac{\mu}{r}} = \frac{1}{r}e^{-\beta}. \tag{18}$$

β can be obtained by rearranging (18):

$$\beta = -\ln\left(2\mu r - 2\sqrt{\mu^2 r^2 - \mu r} - 1\right). \tag{19}$$

Therefore, δ_m can be represented as

$$\delta_m = \frac{1}{r}\left(2\mu r - 2\sqrt{\mu^2 r^2 - \mu r} - 1\right)^{(m-1)}. \tag{20}$$

Let $\theta = (2\mu r - 2\sqrt{\mu^2 r^2 - \mu r} - 1)$. For a positive integer $m \geq 1$, λ_m can be approximated as

$$\lambda_m = m\mu - \sum_{i=1}^{m}\delta_m = m\mu - \frac{1-\theta^m}{r\left(1-\theta\right)}. \tag{21}$$

Consequently, with an anticipated arrival rate λ and the measured service rate μ, the number, denoted as m, of servers that maintain the targeted mean response time r can be approximated as

$$m = \begin{cases} \left\lceil \dfrac{\lambda}{\mu} \right\rceil, & \text{if } \left(\mu\left\lceil \dfrac{\lambda}{\mu} \right\rceil - \dfrac{1-\theta^{\lceil \lambda/\mu \rceil}}{r\left(1-\theta\right)}\right) > \lambda \\ \left\lceil \dfrac{\lambda}{\mu} \right\rceil + 1, & \text{otherwise.} \end{cases} \tag{22}$$

2.2. Approximation for Sizing M/G/m Modeled Clusters. Internet workload characterization has found that the probability of service times is not an exponential distribution but a heavy-tailed distribution in real web systems [12, 14, 30]. In other words, a single-queue m-server cluster should be referred to as $M/G/m$ queue for Internet services. There are no exact formulas for the mean response time of $M/G/m$ system, but numerous approximations can be used. *Kingman's Exponential Law of Congestion* is a popular approximation that is calculated using the coefficient of variation of service times and known solutions from $M/M/m$ queues. Kingman's approximation is expressed as

$$E\left[W^+\right] \approx \frac{1+C^2}{2}E\left[W\right]. \tag{23}$$

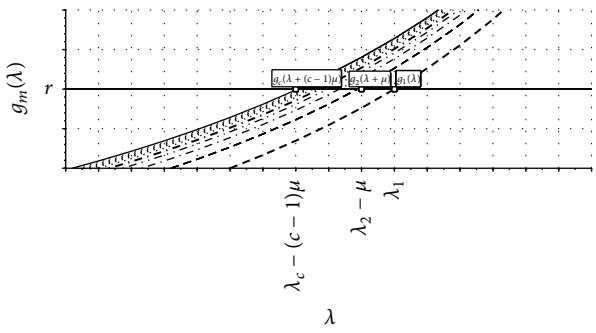


FIGURE 2: $g_m(\lambda)$ versus request arrival rate (λ) with a targeted response time r.

Let $\nu = (1+C^2)/2$. The mean response time of $M/G/m$ system can be expressed as

$$E\left[R^+\right] = \frac{1}{\mu} + E\left[W^+\right] = \frac{1}{\mu} + \nu\frac{p_m}{m\mu\left(1-\lambda/m\mu\right)^2}. \quad (24)$$

Let $f_m^+(\lambda)$ represent the mean response time of $M/G/m$ system on different arrival rates. Based on (8), $f_m^+(\lambda)$ can be expressed as

$$f_m^+(\lambda) = f_m(\lambda) + (\nu-1)\frac{p_m}{m\mu\left(1-\lambda/m\mu\right)^2}. \quad (25)$$

Although $f_m^+(\lambda)$ rises at a more precipitous rate than $f_m(\lambda)$, the correlation observed in Figure 2 and aforementioned approximation still remain valid.

Let the variables $\{\lambda_1^+, \lambda_2^+, \ldots, \lambda_c^+\}$, $\{\delta_1^+, \delta_2^+, \ldots, \delta_c^+\}$, and θ^+ be the correspondences in $M/G/m$ model to the variables $\{\lambda_1, \lambda_2, \ldots, \lambda_c\}$, $\{\delta_1, \delta_2, \ldots, \delta_c\}$, and θ previously mentioned in $M/M/m$ model. The mean response time for $M/G/1$ system can be approximated based on the *Pollaczek-Khintchine* transform:

$$E\left[R_1^+\right] \approx \frac{1}{\mu} + \frac{\nu\rho}{\mu\left(1-\rho\right)} = f_1^+\left(\lambda_1^+\right). \quad (26)$$

Suppose that the targeted response time is still r; then

$$r = f_1^+\left(\lambda_1^+\right) = \frac{1}{\mu} + \nu\frac{\lambda_1^+/\mu}{\mu\left(1-\lambda_1^+/\mu\right)},$$
$$\lambda_1^+ = \frac{\mu\left(r\mu-1\right)}{r\mu-1+\nu}. \quad (27)$$

Similar to the process from (14) to (20), the following equations can be derived:

$$\lambda_2^+ = 2\mu\sqrt{\frac{r\mu-1}{r\mu-1+\nu}},$$
$$\delta_1^+ = \mu - \lambda_1^+ = \frac{\mu\nu}{r\mu-1+\nu},$$
$$\delta_2^+ = \frac{2\mu\left(r\mu-1\right)+\mu\nu}{r\mu-1+\nu} - 2\mu\sqrt{\frac{r\mu-1}{r\mu-1+\nu}},$$
$$\theta^+ = \frac{2\left(r\mu-1\right)+\nu-2\sqrt{\left(r\mu-1\right)\left(r\mu-1+\nu\right)}}{\nu},$$
$$\lambda_m^+ = m\mu - \frac{1-\left(\theta^+\right)^m}{r\left(1-\theta^+\right)}. \quad (28)$$

With an anticipated arrival rate λ and the measured service rate μ, the number, denoted as m^+, of servers that is expected to maintain the required mean response time r can be approximated as

$$m^+ = \begin{cases} \left\lceil \frac{\lambda}{\mu} \right\rceil, & \text{if } \left(\mu\left\lceil \frac{\lambda}{\mu} \right\rceil - \frac{1-\left(\theta^+\right)^{\lceil\lambda/\mu\rceil}}{r\left(1-\theta^+\right)}\right) > \lambda \\ \left\lceil \frac{\lambda}{\mu} \right\rceil + 1, & \text{otherwise.} \end{cases} \quad (29)$$

3. Simulation Setup and Evaluation Metric

A cluster managed by a VOVO scheme periodically adjusts the number of active servers that provide the required services. In general, there are several key functional components including the following:

(1) Job queue: the job queue holds the waiting requests. Each request enters the tail of the queue and waits for service in FCFS manner. In this work, all jobs share a common queue.

(2) Workload distributor: the workload distributor retrieves a job from the head of the job queue and distributes the job to an available node.

(3) Cluster sizing unit: this unit decides the number of active servers. The decision may be based on some predefined thresholds of certain resources, for example, CPU utilization, job throughput, and energy usage. In this work, the decision is calculated based on (22) or (29) according to the given arrival rate, mean service rate, and targeted response time.

(4) On/off controller: the on/off controller periodically activates or deactivates server nodes according to the number given by the sizing unit.

(5) Managed servers: the cluster consists of a group of identical computer nodes, which may be commodity servers. Each server node processes the assigned jobs and reports its working status to the workload distributor.

3.1. The Design of Simulation Program. A simulation program for the VOVO managed system is developed to investigate the performance of the proposed sizing methods. This program is written using the C++ programming language.

In a real VOVO managed system, every incoming job is queued, and an event notification is issued to the workload

```
Simulation(queue, control_interval, state)
cur_period ← −1
next_period ← −1
while queue is not empty do
   Retrieve the job in the head of queue and remove it from queue
   if current_period < 0 then
      cur_period ← job.arrival_time − mod(job.arrival_time, control_interval)
      next_period ← cur_period + control_interval
   else if job.arrival_time ≥ next_period then
      while job.arrival_time ≥ next_period do
         perform statistic of the activated server nodes
         perform cluster sizing
         do on-off control
         cur_period ← next_period
         next_period ← cur_period + control_interval
      end while
   end if
   find an activated server node with the earliest available time
   calculate the waiting time of this job
   calculate the service time of this job
   calculate the response time of this job
   calculate the consumed energy of the server node
end while
```

Algorithm 1: Simulation process.

distributor upon the arrival of a job. If there are available nodes, the workload distributor then dispatches the queued jobs to the available nodes. If a node has completed its assigned job, it also sends an event notification to inform the distributor about its availability. The instructions of node activation and deactivation are periodically issued by the on/off controller. If a deactivation command is issued to a busy node, the node will complete the processing job before turning itself off. However, it will be extremely time consuming to simulate the system with time-based event-driven process. Since the input workload traces have to be readily prepared for the simulation, this work adopts the sequential process that significantly reduces the simulation time. The simulation process is shown in Algorithm 1.

3.2. Randomly Generated Traces. A set of randomly generated traces and two real-world traces are simulated in this work. The most widely used heavy-tailed distribution as the service time distribution is the Pareto distribution [31]. The Poisson distribution is appropriate if the arrivals are from a large number of independent sources, such as web requests [10, 32]. Therefore, the randomly generated traces have Pareto distributed service times with tail indexes from 0.1 to 4.0 stepping by 0.1 and exponentially distributed arrival intervals with traffic intensities from 0.05 to 0.95 stepping by 0.05.

A randomly generated trace T with a tail index l and a traffic intensity ρ is represented by $T_{l,\rho}$. $T_{l,\rho}$ is a series of pairs of an arrival time, denoted by t_a, and a service time, denoted by t_s. Suppose that $T_{l,\rho}$ has n elements; it can be represented as

$$T_{l,\rho} = \{(t_{a1}, t_{s1}), (t_{a2}, t_{s2}), \ldots, (t_{an}, t_{sn})\}. \quad (30)$$

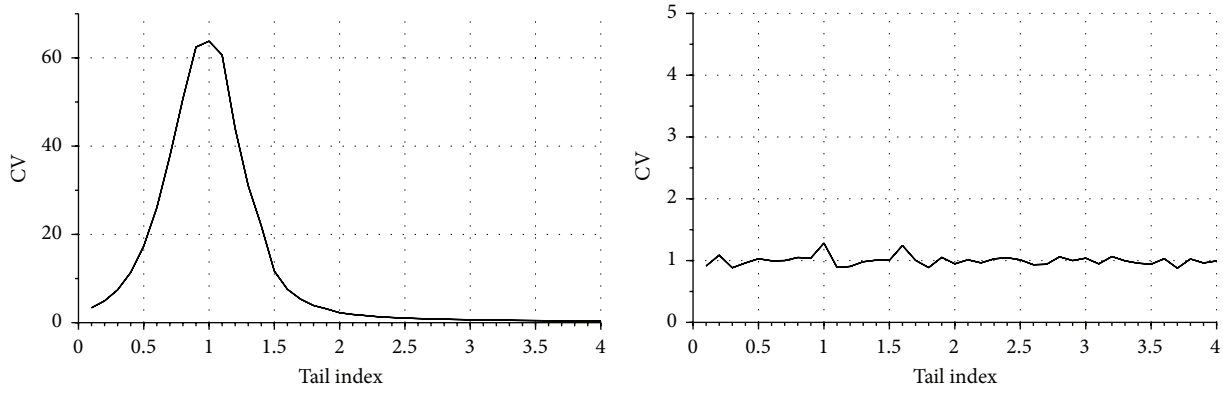
Each unique combination of l and ρ is randomly generated 10 times. That is, there are 10 different traces for a combination of l and ρ. Each trace contains values covering 36,000 time units. All traces are generated with the same mean service time. Therefore, there are 7,600 randomly generated traces which have been simulated in this study. The generating functions for Pareto distributed values and exponential distributed values can be found in many textbooks, for example, [10, p. 509]. The coefficient of variation is often used to measure the relative variation in the data and is the ratio of the standard deviation to the mean. For Pareto distributed values, the coefficient of variation denoted by $\mathrm{CV}_{\mathrm{Pareto}}$ of $l > 2$ can be calculated as [10]

$$\mathrm{CV}_{\mathrm{Pareto}} = \frac{\sqrt{l/(l-1)^2(l-2)}}{l/(l-1)} = \sqrt{\frac{1}{l(l-2)}}, \quad \text{for } l > 2. \quad (31)$$

The coefficient of variation of exponential distributed values is supposed to be 1. The coefficients of variation of service times and arrival intervals of the generated traces are shown in Figures 3(a) and 3(b), respectively.

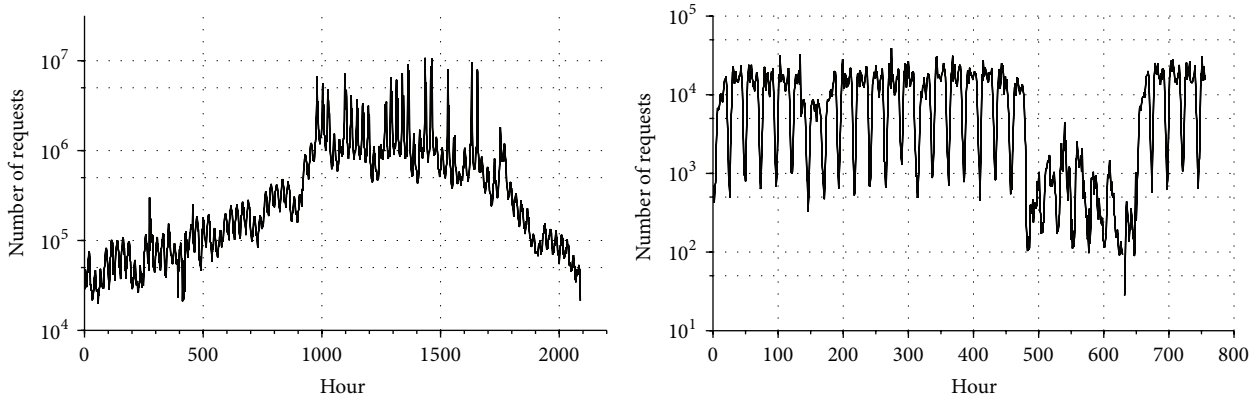
3.3. Real-World Traces. This simulation adopts two real-world workload traces that include a publicly available trace and a trace acquired from a university campus. The service time of a request is assumed to be proportional to its responded page size in the simulation.

The publicly available trace was recorded at the 1998 World Cup web site [30]. This workload trace is one of few



(a) CV of service times versus tail index (b) CV of arrival intervals versus tail index

Figure 3: The coefficients of variation (CV) of service times and arrival intervals of the generated traces.



(a) Traffic pattern of the World Cup log (b) Traffic pattern of the campus log

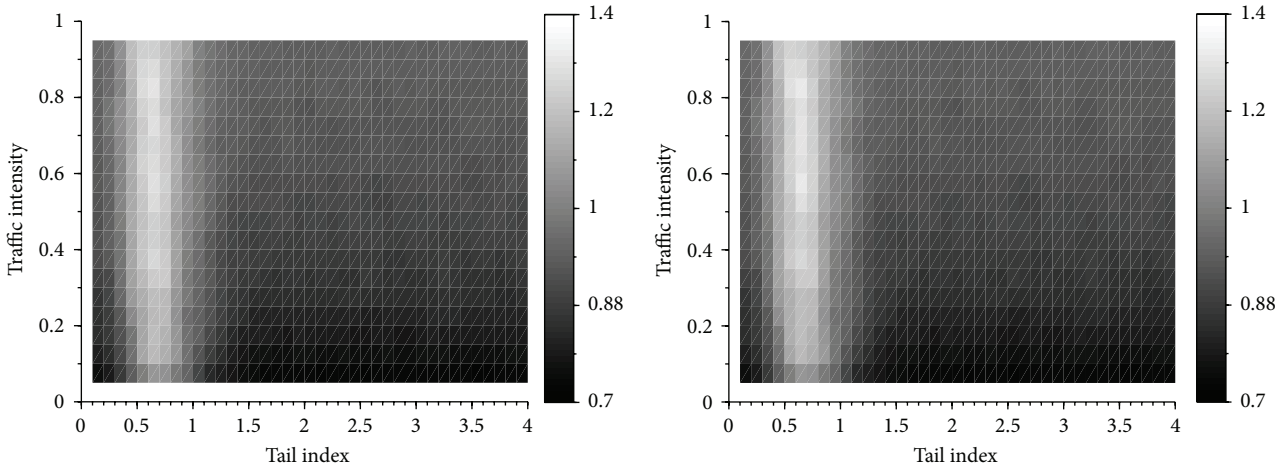
Figure 4: Hourly traffic patterns of the adopted real-world web traces.

logs providing server activation records. It is known for having a heavy-tailed page-size distribution with a tail index of 1.37 [30]. Each request recorded in the log contains an arrival time, a responded page size, and a server identification. The 1998 World Cup log was collected from 05:30:17 May 1, 1998, through 05:59:55 July 27, 1998, a total of 87 days. The log exhibits the following characteristics: 1,352,804,107 requests, 33 hosting servers, 4,040.684 bytes per response in average, 108.71 requests per second per server (the peak service rate) [30], and an average service time of 0.0092 seconds per request with a standard deviation of 0.084.

The second workload trace is acquired from a university with a student population of 4,219, including 3,531 undergraduates. This web access log was collected from 12:03:59 September 19, 2014, through 00:01:39 October 21, 2014, a total of 31 days. The trace log is from a site hosting a student information system that provides course information, handouts/homework systems, message system, email system, and other campus information. The log exhibits the following characteristics: 7,054,170 requests, 8 hosting servers, 5,991.64 bytes per response in average, 74.74 requests per second per server (the peak service rate), an average service time of 0.0134 seconds per request with a standard deviation of 0.227, and a tail index of 0.154 of the service time distribution.

The hourly traffic patterns of the 1998 World Cup log and the 2014 campus log are shown in Figures 4(a) and 4(b), respectively. The two logs represent two distinct service patterns including an occasional service pattern, that is, 1998 World Cup, and a regular service pattern, that is, student information system. The World Cup log shows a growth-decay pattern. An iterative pattern analogous to the daily working hours is observed in the campus log. Note that there are a school break and a scheduled maintenance during the recorded period.

For the World Cup log, the simulated cluster consists of 33 servers based on the information given in the log. For the campus log, the simulated cluster consists of 8 servers. As for the randomly generated traces, the simulated cluster consists of 10 servers. The on/off controller periodically sizes the simulated cluster with the interval set at 300 seconds, which are long enough to compensate the machine boot-up delays and short enough to reflect the demand changes [1, 2, 33].



(a) Relative performance of *mgm* scenario

(b) Relatively performance of *mmm* scenario

FIGURE 5: Relative performance of the randomly generated traces with $\omega_S = 1$, $\omega_A = 1$, and $\omega_E = 1$.

3.4. Evaluation Metric. Three simulation scenarios, which are all-on, *mmm*, and *mgm*, are performed. All servers in a cluster are always powered on in all-on scenario. This scenario is expected to consume the most energy but to have the best service quality. The *mmm* scenario uses (22) to approximate the number of servers. The *mgm* scenario is similar to *mmm* except that (29) is used for the sizing approximation. Nielsen's [34] response time limits for usability are adopted by setting the targeted response time at 1 second and the failure threshold at 10 seconds.

The objective of a VOVO scheme is to reduce the energy consumption while maintaining a reasonable service quality. To gauge the performance of an approach x (denoted by η_x), relative measures to all-on are adopted instead of absolute measurements, since the all-on scenario must have the least response time and the highest energy consumption. The considered factors of a scenario x are as follows:

(1) being satisfactory, denoted by S_x, which is the portion of responses conforming to the targeted response time;

(2) acceptance, denoted by A_x, which is the portion of responses being admissible (i.e., under the failure threshold);

(3) energy, denoted by E_x, which is the average number of activated servers, since all servers are identical and have the same power profile.

The relative measurements of S_x, A_x, and E_x are defined as

$$S_x^+ = \frac{S_{\text{all-on}}}{S_x}, \quad \text{for } S_x > 0,$$
$$A_x^+ = \frac{A_{\text{all-on}}}{A_x}, \quad \text{for } A_x > 0, \tag{32}$$
$$E_x^+ = \frac{E_x}{E_{\text{all-on}}}, \quad \text{for } E_{\text{all-on}} > 0.$$

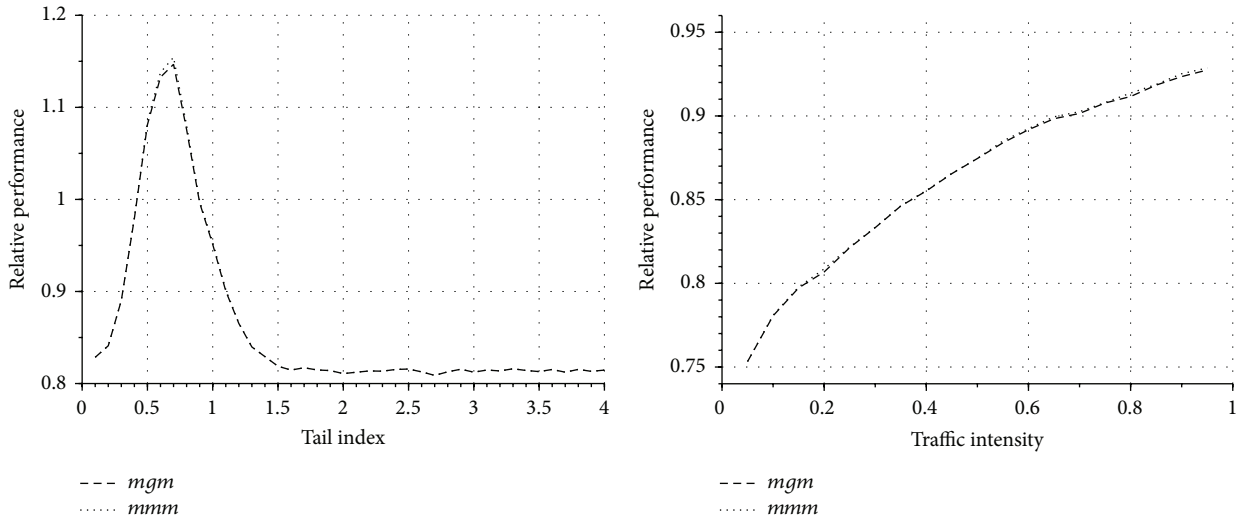
Let ω_S, ω_A, and ω_E be the weighting coefficients for S_x^+, A_x^+, and E_x^+, respectively. The relative performance, denoted by η_x, is defined as

$$\eta_x = \frac{\left(S_x^+\omega_S + A_x^+\omega_A + E_x^+\omega_E\right)}{\left(\omega_S + \omega_A + \omega_E\right)}. \tag{33}$$

4. Simulation Results and Analysis

4.1. Simulation Results. With this relative measurement, that is, (33), the optimal solution produces the minimal value of η_x. The simulation results of randomly generated traces are summarized by the relative performance of the simulated scenarios to all-on with $\omega_S = 1$, $\omega_A = 1$, and $\omega_E = 1$. In order to make the results be easily comprehended, the relative performances of η_{mgm} and η_{mmm} are graphically visualized using gray level. Figure 5 shows the relative performance, that is, η_x, of scenarios *mgm* and *mmm*, with $\omega_S = 1$, $\omega_A = 1$, and $\omega_E = 1$. It is very difficult to visually differentiate Figures 5(a) and 5(b). Using the averaged values, as shown in Figure 6, it can be found that *mgm* has a slightly better performance than *mmm*. In average, which is based on Figure 6, scenarios *mgm* and *mmm* outperform all-on under most cases except when the tail index is between 0.4 and 0.9. Furthermore, the averaged relative performances shown in Figure 6(a) are clearly correlated with the coefficient of variation of service times (as shown in Figure 3(a)). This simulation result indicates that both *mgm* and *mmm* yield a worse performance than all-on for diverse access patterns. This may imply that these approaches undersize the cluster for high variation of service times.

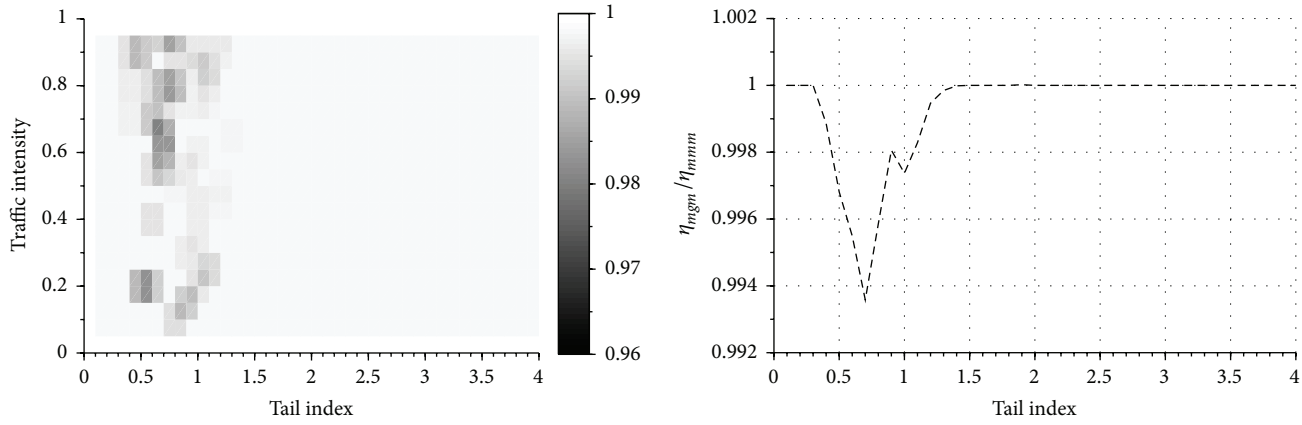
In Figures 6(a) and 6(b), the curves of *mgm* and *mmm* are indistinguishable under those scales. In fact, the relative performances of scenarios *mgm* and *mmm* are not identical. Figure 7(a) shows the ratios of η_{mgm} to η_{mmm}. There are some regions between tail indexes 0.3 and 1.3 where the ratios are not 1, that is, identical. In Figure 7(b), the average of η_{mgm}/η_{mmm} is always less than or equal to 1, which means that



(a) Relative performances versus tail index

(b) Relative performances versus traffic intensity

Figure 6: Average relative performance of the randomly generated traces with $\omega_S = 1$, $\omega_A = 1$, and $\omega_E = 3$.



(a) Relative performance ratios of *mgm* to *mmm*

(b) Average of relative performance ratios

Figure 7: The relative performance ratios of η_{mgm}/η_{mmm} with $\omega_S = 1$, $\omega_A = 1$, and $\omega_E = 1$.

$M/G/m$ based sizing is more effective than $M/M/m$ based sizing. However, Figure 7 also shows that the difference is very small, that is, under 1% in average. Given the fluctuation nature of web traffic, $M/M/m$ based sizing may be adequate for empirical practices.

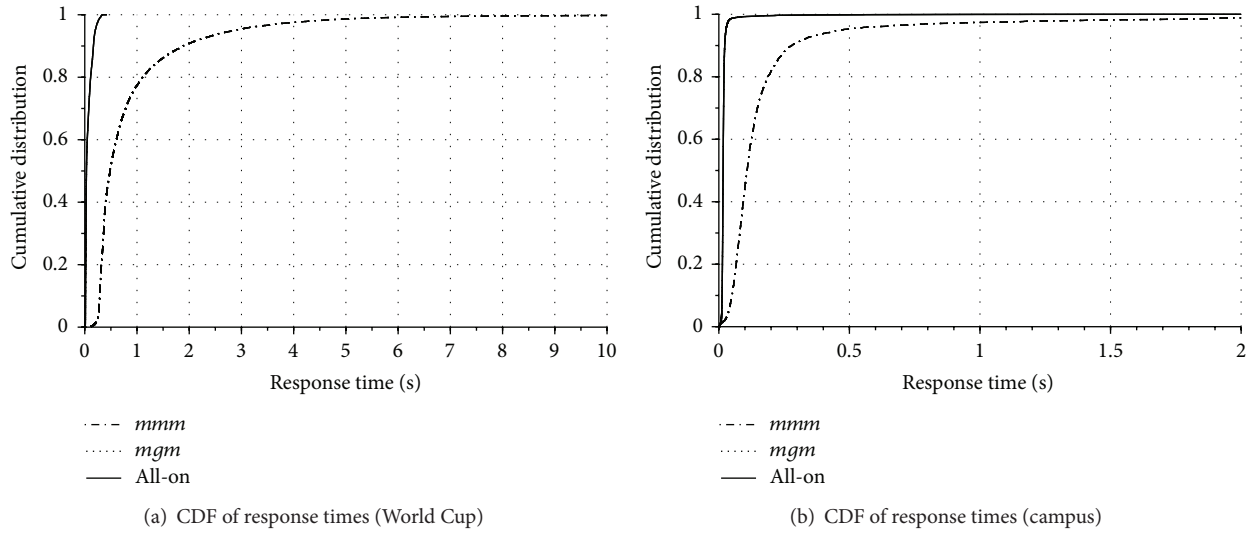
In order to examine above findings, two real-world traces are simulated under previously mentioned scenarios, that is, all-on, *mgm*, and *mmm*. Figure 8 shows the cumulative distribution of the response times of the simulated real-world traces. As shown in Figure 8(a), all requests in scenario all-on can be served within 1 second, but only approximately 80% of requests can be handled for this targeted response time in scenarios *mgm* and *mmm*. The curves of *mgm* and *mmm* are also indistinguishable in Figure 8(a). In Figure 8(b), more than 99.96% of requests in scenario all-on can be served within 1 second. More than 97% of requests can be handled for this targeted response time in *mgm* and *mgm* scenarios. The curves of *mgm* and *mmm* are also indistinguishable in Figure 8(b).

Based on the relative performance, that is, η_x, Table 1 shows that *mmm* and *mgm* are very similar in both cases. As expected, all-on always has the shortest mean response time but the most energy consumption. The proposed queueing-based sizing approaches, that is, *mmm* and *mgm*, can reduce significant energy consumption while maintaining a reasonable service quality.

4.2. Analysis and Comparison. Energy consumption and service quality of the server machines are two major performance measures for a cloud service provider. The above results are fully based on simulation. To evaluate the proposed strategy on a real system, a 6-hour log is extracted from the World Cup trace and fed to a cluster consisting of 33 computers. In addition to the 33-node cluster, there

Table 1: Relative performance.

Log	x	S_x	A_x	E_x	S_x^+	A_x^+	E_x^+	η_x
World Cup	All-on	1.000	1.000	33	1.000	1.000	1.000	1.000
	mmm	0.817	0.993	2.389	1.224	1.007	0.072	0.768
	mgm	0.817	0.993	2.390	1.223	1.007	0.072	0.767
Campus	All-on	0.999	0.999	10	1.000	1.000	1.000	1.000
	mmm	0.974	0.998	1.32	1.026	1.002	0.132	0.720
	mgm	0.974	0.998	1.32	1.026	1.002	0.132	0.720



(a) CDF of response times (World Cup)

(b) CDF of response times (campus)

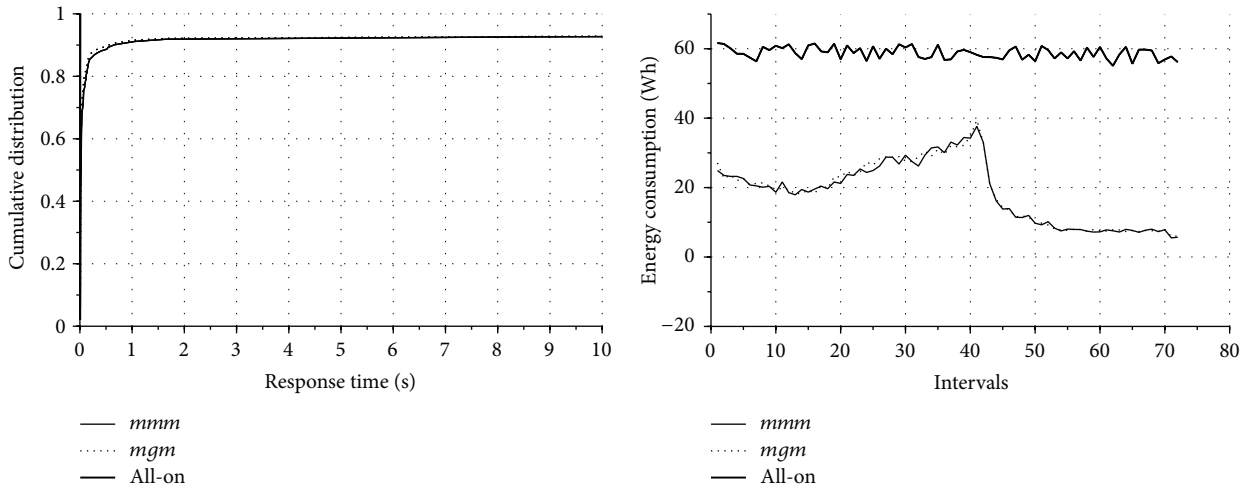
Figure 8: Cumulative distribution of the response times of two real-world traces.

are an external computer that hosts other key functional components mentioned in Section 3 and a network switch connecting all nodes and the external computer through 1000BASE-T Ethernet. The extracted log contains 22,821,177 access records, which are from June 29, 1998, 17:20:00 GMT to June 29, 1998, 23:19:59 GMT. Each node of the cluster is equipped with a dual-core 1.66 GHz Intel Atom N280 processor and 1 GB of memory. All nodes use Linux 2.6 as the operation system with Apache 2.2 installed. The average power demand is 20.83 Watts when an idle node waits for a request with all its parts being turned on. The peak power level of a node that was instrumented is 26.33 Watts. The node profile of the test cluster is shown in Table 2.

In the evaluation, the on/off controller periodically sizes the cluster with the interval set at 300 seconds. Interval energy data of the cluster, excluding the external computer and the network switch, is instrumented and stored by a digital multimeter (DMM). The evaluation result is shown in Figure 9 and conforms to the simulation results. As shown in Figure 9(a), with all nodes turned on, that is, all-on scenario, all requests are responded to within 1 second, while only approximately 92% of requests can be responded to within 1 second for either *mmm* scenario or *mgm* scenario. On the other hand, both *mmm* scenario and *mgm* scenario consume much less energy than all-on scenario, as shown in Figure 9(b). Similar to the simulation results, the curves of *mgm* and *mmm* are also very close to each other in both Figures 9(a) and 9(b).

VOVO strategy has been studied for more than a decade. Many VOVO approaches [33, 35–37], which dynamically size a cluster according to a preset threshold of CPU utilization or resource usage, were developed based on the designs proposed by Chase et al. [1] or Pinheiro et al. [2]. To compare the proposed queueing-based approach with the threshold-based approaches, Pinheiro's approach [2] is simulated and denoted as *vovo* scenario. In *vovo*, the service demand is smoothed and estimated using the cumulative moving average. *vovo* periodically activates one more node of the cluster when the estimated utilization rate exceeds a predefined threshold and deactivates one node otherwise. The World Cup trace is also used in the simulation of *vovo*. Since *vovo* uses the threshold of CPU utilization rate instead of the response time as a controlling factor, 3 different threshold values, which are 0.7, 0.8, and 0.9, are simulated to get a comparable result.

The simulation results of *vovo* are evaluated with the metric proposed in Section 3.4 and compared with all-on and *mmm*, as shown in Table 3. From this comparison, the threshold of the CPU utilization rate has to be less than 0.8 for *vovo* to get a comparable result with *mmm*. Although *vovo* outperforms *mmm* with the threshold set at 0.7, it requires more nodes and therefore consumes more energy than *mmm*. In order to get a reasonable threshold value for *vovo*, it may be necessary to go through several runs of simulation or other lengthy procedures. On the other hand, the proposed approach minimally requires only the anticipated arrival rate



(a) Response times (b) Energy usage

Figure 9: Real system evaluation.

Table 2: Node profile of the test cluster.

Model	Quantity	Processor	Disk
Acer Veriton N260G	33	1.66 GHz Intel Atom N280	320 GB SATA2 Hitachi HTS54503
Memory	OS	Idle Power	Peak Power
1 GB	Linux 2.6	20.83 W	26.33 W

Table 3: Performance comparison of all-on, *mmm*, and *vovo*.

Log	x	S_x	A_x	E_x	S_x^+	A_x^+	E_x^+	η_x
World Cup	All-on	1.000	1.000	33	1.000	1.000	1.000	1.000
	mmm	0.817	0.993	2.389	1.224	1.007	0.072	0.768
	vovo(0.9)	0.662	0.842	2.162	1.509	1.187	0.066	0.921
	vovo(0.8)	0.812	0.927	2.382	1.231	1.078	0.072	0.794
	vovo(0.7)	0.904	0.970	2.661	1.105	1.030	0.081	0.739

λ, the service rate μ, and the desired response time r to approximate the required number of servers m, that is, (22).

5. Conclusion

This paper proposes two queueing-based sizing methods to periodically adjust the number of servers in a cluster. The proposed method aims at achieving a fair energy-delay performance trade-off of server clusters. The proposed approximation formulas, that is, (22) and (29), are simple closed-form expressions, which may be implemented in a network switch for real-time processing.

From the simulation results, the schemes with the proposed approximation formulas reduce considerable amount of energy consumption while maintaining comparable service performance for gentle service time fluctuations. However, the proposed methods tend to underestimate the number of required servers for service processes with high variability, that is, tail index between 0.3 and 1.3. Similar observation has also been documented in [5].

The relative measurements of *mmm* and *mgm* are almost undifferentiated, except that *mgm* is very slightly better than *mmm* for service processes with high variability. Although Internet workload characterization has found that the probability of service times is a heavy-tailed distribution, periodically resizing the cluster is possible to alleviate the situation of long jobs blocking short jobs in the waiting queue. Because once a deactivation command is issued to a busy node, the node becomes a pending-off node that has to complete the unfinished job before turning itself off. If a long job is handled by this pending-off node, the queued jobs can be quickly assigned to other newly activated nodes of the next period without waiting for the finish of that long job. Therefore, sizing the cluster based on $M/M/m$ model or $M/G/m$ model makes little difference. Based on the simulation results, the simpler $M/M/m$ model may be adequate and preferable for sizing clusters adopting queueing-based VOVO schemes.

Server clusters are widely adopted in cloud data centers [28]. In order to support various kinds of services including user-end applications and back-end activities, heterogeneity becomes common in multifunctional cloud data centers. It is popular that a data center has different group of servers with different computation capacities. Since the basic computing units that are grouped for specific function usually consist of the same type of machines, the proposed approach is built based on the assumption of homogeneous nodes. Therefore, the proposed approach is particularly pertinent for the computing units forming the underling base of cloud data centers. Nevertheless, extending this work to the heterogeneous environments is an immediate future work of this study. The multitier system is an obvious case of server heterogeneity and is widely adopted in many enterprise systems. There are many approaches which have been proposed to address the applicability of queueing models on multitier systems, such as Multitier Internet Applications [25], Heterogeneous Multitier Web Clusters [38], Layered Queueing Networks (LQN) [39, 40], and Power-Saving Server Farms [41]. The job dispatching [42, 43] and scheduling [44, 45] also arise as important issues in a heterogeneous environment. Considering these related developments and integrating the proposed approach with the existing work may be a practical way to extend this study to a heterogeneous environment.

Symbols and Definitions

λ: The job arrival rate of a queueing system
μ: The mean service rate of a server in a queueing system
$1/\mu$: The mean service time of a server in a queueing system
σ: The standard deviation of the service times in a queueing system
m: The number of servers in a queueing system
ρ: The traffic intensity, $\rho = \lambda/m\mu$
j: A system state, which is the same as the number of jobs in the system
p_j: The probability of a state j
C: The coefficient of variation of service times in a queueing system, $C = \sigma\mu$
N: The number of jobs in $M/M/m$ system
N_1: The number of jobs in $M/M/1$ system
M: The number of busy servers in $M/M/m$ system
$E[N]$: The mean value of N
$E[N_1]$: The mean value of N_1
$E[M]$: The mean value of M
R: The response time of a job in $M/M/m$ system
R_1: The response time of a job in $M/M/1$ system
R^+: The response time of a job in $M/G/m$ system
R_1^+: The response time of a job in $M/G/1$ system
W: The waiting time of a job in $M/M/m$ system
W^+: The waiting time of a job in $M/G/m$ system
$E[R]$: The mean value of R
$E[R_1]$: The mean value of R_1
$E[R^+]$: The mean value of R^+
$E[R_1^+]$: The mean value of R_1^+
$E[W]$: The mean value of W
$E[W^+]$: The mean value of W^+.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This study is funded by the Ministry of Science and Technology (Taiwan) under Grant no. NSC 101-2632-E-036-001-MY3 for the project *A Study of Applications and Examinations on the Smart Meter Enabled Electricity Grid*.

References

[1] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle, "Managing energy and server resources in hosting centers," *SIGOPS—Operating Systems Review*, vol. 35, pp. 103–116, 2001.

[2] E. Pinheiro, R. Bianchini, E. Carrera, and T. Heath, "Load balancing and unbalancing for power and performance in cluster-based systems," in *Proceedings of the Workshop on Compilers and Operating Systems for Low Power (COLP '01)*, vol. 180, pp. 182–195, Barcelona, Spain, 2001.

[3] E. Brockmeyer, H. L. Halstrm, A. K. Erlang, and A. Jensen, *The Life and Works of A.K. Erlang*, Transactions of the Danish Academy of Technical Sciences, Akademiet for de Tekniske Videnskaber, 1948.

[4] R. Guerra, L. Bertini, and J. Leite, "Improving response time and energy efficiency in server clusters," in *Proceedings of the 8th Workshop de Tempo*, p. 8, Curitiba, Brazil, May 2006.

[5] D. Meisner, B. T. Gold, and T. F. Wenisch, "PowerNap: eliminating server idle power," *ACM SIGPLAN Notices*, vol. 44, no. 3, pp. 205–216, 2009.

[6] X. Zheng and Y. Cai, "Markov model based power management in server clusters," in *Proceedings of the 2010 IEEE/ACM International Conference on Green Computing and Communications and International Conference on Cyber, Physical and Social Computing (CPSCom '10)*, pp. 96–102, Washington, DC, USA, 2010.

[7] R. Buyya, A. Beloglazov, and J. Abawajy, "Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges," in *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '10)*, pp. 6–17, CSREA Press, 2010.

[8] A. Gandhi, V. Gupta, M. Harchol-Balter, and M. A. Kozuch, "Optimality analysis of energy-performance trade-off for server farm management," *Performance Evaluation*, vol. 67, no. 11, pp. 1155–1171, 2010.

[9] A. Gandhi, M. Harchol-Balter, R. Raghunathan, and M. A. Kozuch, "Autoscale: dynamic, robust capacity management for multi-tier data centers," *ACM Transactions on Computer Systems*, vol. 30, no. 4, article 14, 2012.

[10] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*, John Wiley & Sons, 1991.

[11] M. Harchol-Balter and A. B. Downey, "Exploiting process lifetime distributions for dynamic load balancing," *ACM Transactions on Computer Systems*, vol. 15, no. 3, pp. 253–285, 1997.

[12] M. E. Crovella, M. S. Taqqu, and A. Bestavros, "Heavy-tailed probability distributions in the world wide web," in *A Practical Guide to Heavy Tails: Statistical Techniques and Applications*, R. J. Adler, R. E. Feldman, and M. S. Taqqu, Eds., pp. 3–25, Birkhäuser, Boston, Mass, USA, 1998.

[13] A. Williams, M. Arlitt, C. Williamson, and K. Barker, "Web workload characterization: ten years later," in *Web Content Delivery*, X. Tang, J. Xu, and S. Chanson, Eds., vol. 2 of *Web Information Systems Engineering and Internet Technologies Book Series*, pp. 3–21, Springer, New York, NY, USA, 2005.

[14] D. Ersoz, M. S. Yousif, and C. R. Das, "Characterizing network traffic in a cluster-based, multi-tier data center," in *Proceedings of the 27th International Conference on Distributed Computing Systems (ICDCS '07)*, p. 59, IEEE, Toronto, Canada, June 2007.

[15] S. Mirtchev and R. Goleva, "Discrete time single server queueing model whit a multimodal packet size distribution," in *Proceedings of the Conjoint Seminar on Modeling and Control of Information Processes*, T. Atanasova, Ed., pp. 83–101, Sofia, Bulgaria, 2009.

[16] M. J. Fischer, D. M. B. Masi, D. Gross, and J. F. Shortle, "One-parameter pareto, two-parameter pareto, three-parameter pareto: is there a modeling difference?" *Alcatel Telecommunications Review*, pp. 79–92, 2005.

[17] A. Gandhi and M. Harchol-Balter, "How data center size impacts the effectiveness of dynamic power management," in *Proceedings of the 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton '11)*, pp. 1164–1169, September 2011.

[18] D. Meisner, B. T. Gold, and T. F. Wenisch, "The powernap server architecture," *ACM Transactions on Computer Systems*, vol. 29, no. 1, article 3, 2011.

[19] H. Goudarzi, M. Ghasemazar, and M. Pedram, "SLA-based optimization of power and migration cost in cloud computing," in *Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid '12)*, pp. 172–179, May 2012.

[20] Z. Liu, Y. Chen, C. Bash et al., "Renewable and cooling aware workload management for sustainable data centers," in *Proceedings of the 12th ACM SIGMETRICS/Performance Joint International Conference on Measurement and Modeling of Computer Systems*, pp. 175–186, June 2012.

[21] A. Gandhi, S. Doroudi, M. Harchol-Balter, and A. Scheller-Wolf, "Exact analysis of the M/M/k/setup class of Markov chains via recursive renewal reward," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 1, pp. 153–166, 2013.

[22] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam, "Managing server energy and operational costs in hosting centers," *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, pp. 303–314, 2005.

[23] D. Meisner, C. M. Sadler, L. A. Barroso, W. Weber, and T. F. Wenisch, "Power management of online data-intensive services," in *Proceeding of the 38th Annual International Symposium on Computer Architecture*, pp. 319–330, San Jose, Calif, USA, June 2011.

[24] Y. Zhang, Y. Wang, and X. Wang, "Electricity bill capping for cloud-scale data centers that impact the power markets," in *Proceedings of the 41st International Conference on Parallel Processing (ICPP '12)*, pp. 440–449, September 2012.

[25] B. Urgaonkar, P. Shenoy, A. Chandra, and P. Goyal, "Dynamic provisioning of multi-tier internet applications," in *Proceedings of the 2nd International Conference on Autonomic Computing (ICAC '05)*, pp. 217–228, June 2005.

[26] V. Gupta, M. Harchol-Balter, J. G. Dai, and B. Zwart, "On the inapproximability of *M/G/K*: why two moments of job size distribution are not enough," *Queueing Systems*, vol. 64, no. 1, pp. 5–48, 2010.

[27] D. Meisner and T. F. Wenisch, "Stochastic queuing simulation for data center workloads," in *Proceedings of the Exascale Evaluation and Research Techniques Workshop*, p. 9, March 2010.

[28] X. Liao, L. Hu, and H. Jin, "Energy optimization schemes in cluster with virtual machines," *Cluster Computing*, vol. 13, no. 2, pp. 113–126, 2010.

[29] K. S. Trivedi, *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*, Wiley-Interscience, 2nd edition, 2001.

[30] M. Arlitt and T. Jin, "Workload characterization study of the 1998 world cup web site," *IEEE Network*, vol. 14, no. 3, pp. 30–37, 2000.

[31] Z. Tari, A. K. A. Phan, M. Jayasinghe, and V. G. Abhaya, *On the Performance of Web Services*, Springer, 2011.

[32] H. Gupta, A. Mahanti, and V. J. Ribeiro, "Revisiting coexistence of poissonity and self-similarity in internet traffic," in *Proceedings of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS '09)*, pp. 1–10, London , UK, September 2009.

[33] E. N. Elnozahy, M. Kistler, and R. Rajamony, "Energy-Efficient Server Clusters," in *Power-Aware Computer Systems*, B. Falsafi and T. Vijaykumar, Eds., vol. 2325 of *Lecture Notes in Computer Science*, pp. 179–197, Springer, Berlin, Germany, 2003.

[34] J. Nielsen, *Usability Engineering*, Morgan Kaufmann Publishers, 1993.

[35] W. Chen, F. Jiang, W. Zheng, and P. Zhang, "A dynamic energy conservation scheme for clusters in computing centers," in *Embedded Software and Systems*, vol. 3820 of *Lecture Notes in Computer Science*, pp. 244–255, Springer, Berlin, Germany, 2005.

[36] X. Zheng and Y. Cai, "Optimal server provisioning and frequency adjustment in server clusters," in *Proceedings of the 39th International Conference on Parallel Processing Workshops (ICPPW '10)*, pp. 504–511, IEEE, San Diego, Calif, USA, September 2010.

[37] W. Wei, L. Junzhou, S. Aibo, and D. Fang, "Energy-aware dynamic server provisioning and frequency adjustment in multi-tier data centers," *Journal of Internet Technology*, vol. 14, no. 4, pp. 609–618, 2013.

[38] P. Wang, Y. Qi, X. Liu, Y. Chen, and X. Zhong, "Power management in heterogeneous multi-tier web clusters," in *Proceedings of the 39th International Conference on Parallel Processing (ICPP '10)*, pp. 385–394, IEEE, San Diego, Calif, USA, September 2010.

[39] G. Franks, P. Maly, M. Woodside, D. C. Petriu, and A. Hubbard, "Layered queueing network solver and simulator user manual," Tech. Rep., Department of Systems and Computer Engineering, Carleton University, 2005.

[40] Y. Shoaib and O. Das, "Web application performance modeling using layered queueing networks," *Electronic Notes in Theoretical Computer Science*, vol. 275, no. 1, pp. 123–142, 2011.

[41] S. Wang, W. Munawar, X. Liu, and J.-J. Chen, "Power-saving design in server farms for multi-tier applications under

response time constraint," in *Proceedings of the 2nd International Conference on Smart Grids and Green IT Systems (SMARTGREENS '13)*, pp. 137–148, May 2013.

[42] V. Gupta, *Stochastic models and analysis for resource management in server farms [Ph.D. thesis]*, Intel Corporation, 2011.

[43] C.-J. Tang, M.-R. Dai, C.-C. Chuang, Y.-S. Chiu, and W. S. Lin, "A load control method for small data centers participating in demand response programs," *Future Generation Computer Systems*, vol. 32, no. 1, pp. 232–245, 2014.

[44] B. Urgaonkar, G. Pacifici, P. Shenoy, M. Spreitzer, and A. Tantawi, "An analytical model for multi-tier internet services and its applications," *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, no. 1, pp. 291–302, 2005.

[45] M. Mazzucco and D. Dyachuk, "Balancing electricity bill and performance in server farms with setup costs," *Future Generation Computer Systems*, vol. 28, no. 2, pp. 415–426, 2012.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 928193, 7 pages
http://dx.doi.org/10.1155/2015/928193

Research Article

A Novel Trust-Aware Composite Semantic Web Service Selection Approach

Denghui Wang,[1] Hao Huang,[2] and Changsheng Xie[3]

[1]*School of Computer Science and Technology, Huazhong University of Science & Technology, Wuhan 430074, China*
[2]*School of Software Engineering, Huazhong University of Science & Technology, Wuhan 430074, China*
[3]*Wuhan National Laboratory for Optoelectronics, Wuhan 430074, China*

Correspondence should be addressed to Hao Huang; thao@hust.edu.cn

Received 19 March 2015; Revised 12 June 2015; Accepted 14 June 2015

Academic Editor: Bao Rong Chang

The functional characteristics and the nonfunctional properties of service profile always play very important roles in composite semantic web service selection approach. But the credibility of this information cannot be guaranteed. This paper established a novel trust degree model of this information. Based on this model, the trust degrees can be calculated from execution log and user experience evaluation of candidate web services. Then the paper proposes a new composite semantic web service selection approach based on this credible information. Finally, we present two experiments to prove that the new approach can avoid the influence of exaggerated and unauthentic information effectively and accurately.

1. Introduction

With the rapid development of sophisticated application, a single web service is usually too simple to meet the various user requirements. Creating new services through web service composition to provide more complex and on-demand functions becomes essential. Web service composition is introduced to resolve the above problem. However, there exist a number of available web services providing similar or identical functional characteristics, so users need a selection approach that can help them choose the best composite web service.

According to user request, the service selection approach should consider two aspects that include features and quality of candidate service. The feature of web service can be described by inputs, outputs, precondition, and effects (IOPE) in semantic web service model. IOPE is called functional property. In many cases, composition techniques and the related tools exploit IOPE predicates that characterize structural and semantic services descriptions to generate the desired compositions [1]. And the QoS represent the quality of web service. QoS is called nonfunctional property. Recently most of researchers use QoS as the main parameters of composite web service selection [2–4]. However, in most of the current web service composition selection methods, it is assumed that the service profile offered by different organization is trusted and authentic. But in fact some organizations of web service are apt to publish unauthentic information for attracting end user. To cope with this problem, it is necessary to avoid exaggerated information by dishonest providers in the selection process. In this paper, we establish a new trust degree model of composite semantic web service. And, based on this model, we propose a novel composite web service selection approach which can calculate the trust degree of QoS and IOPE information from execution log and credible user experience evaluation.

This paper is organized as follows: in Section 2, we emphatically introduce the related work about composite web service selection. Section 3 introduces the trust degree model for getting reliable QoS value and credible IOPE information. Section 4 explains the approach to calculating trust degree of QoS and IOPE for selecting the best composite web service. Section 5 proposes the experiments that lead to proving that the proposed approach is accurate and effective. Section 6 gives the final conclusion.

2. Related Works

The traditional selection and composition of web services rely on the manner of finding the most similar functionalities and the best nonfunctionalities of web service. The QoS is widely employed for describing nonfunctionalities. There are a lot of service selection methods expanded in this regard. Reference [5] proposed an algorithm to combine global QoS constraints with local selection. Reference [6] proposed an approach for web service dynamic composition based on global QoS constraints decomposition. Besides global QoS, [7] proposed a distributed optimal scheme based on local QoS. Reference [8] presented QoS-GRASP, a metaheuristic algorithm for performing QoS-aware web service composition at runtime. QoS-GRASP is a hybrid approach that combines GRASP with Path Relinking. References [9, 10] applied the neurofuzzy decision making approach in the process of selection and choice of the most appropriate web service with respect to quality of service criteria. The method deals with the imprecision of QoS constraints values. And these QoS-based service selection methods always assume that the QoS data coming from service providers and users are effective and trustworthy, which is actually impossible in real environment. So these web service selection methods mentioned above are not perfect. In [11], the authors proposed a novel service composition approach, modeling the trust-based service composition as the multidomain scheduling and assignment problem using the minimum service resources within a certain time constraint. They considered that trust plays a pivotal role in service composition approach. In [12], the researchers presented a trustworthy services selection based on preference selection method that assists users in selecting the right web service, according to their own preference. This method can effectively solve the weaknesses of recommendation systems. In [13], we proposed one method for web service recommendation based on trust-aware QoS. But the method did not consider the functionality property. Thus, this paper based on the above study proposes a novel composite web service selection method according to credible QoS and IOPE information.

3. Trust Degree Model

3.1. Semantic Web Service Model

Definition 1 (semantic web service). A semantic web service is defined as a quadruple:

$$\mathrm{WS} = \langle \mathrm{SN}, \mathrm{Des}, F, \mathrm{QoS} \rangle , \tag{1}$$

where SN is the identifier of the service; Des is the general information about the service, including service contexts such as text description, ontology definition, version, contractor, and the general information being independent of the specific function; F is the functional attribute of a service; QoS is a set of attribute parameters standing for the quality of service, including some attributes such as cost, response time, successful execution rate, and availability. The last two parameters in the quadruple WS are actually related with service composition.

Moreover, semantic web service's function attribute can be demonstrated as a quadruple:

$$F = \langle I, O, P, E \rangle , \tag{2}$$

where I is the set of service's semantic inputs; O is the set of service's semantic outputs; P is the precondition; E is the effects of the semantic web service. Definitely, a semantic web service can be expressed as follows:

$$\mathrm{WS} = \langle \mathrm{SN}, \mathrm{Des}, \langle I, O, P, E \rangle , \mathrm{QoS} \rangle . \tag{3}$$

3.2. User Requirement Model

Definition 2 (user request). A user request of semantic web service can be expressed as a quadruple:

$$\mathrm{WS}_r = \left\langle \mathrm{SN}_r, \mathrm{Des}_r, F_r, \mathrm{QoS}_r \right\rangle ; \quad F_r = \left\langle I_r, O_r, P_r, E_r \right\rangle . \tag{4}$$

The service requester's information is in Des_r. A service request can be expressed as follows:

$$\mathrm{WS}_r = \left\langle \mathrm{SN}_r, \mathrm{Des}_r, \left\langle I_r, O_r, P_r, E_r \right\rangle , \mathrm{QoS}_r \right\rangle . \tag{5}$$

User request model contains several requirement indexes, and the consumer provides the corresponding constraint condition for every index. But the expression of every index is different, so we need to normalize these indexes. The expression of user requirement indexes can be divided into two categories. One is the value type. And the other one is the interval type. In order to facilitate comparison with the real value, we need to transform the interval type to the value type. At present, there is a variety of deterministic methods. In this paper we use the expanded ordered weighted averaging operator as determining the mathematical formula.

Assume $F : R^n \rightarrow R$; if $F(a_1, a_2, \ldots, a_n) = \sum \omega_j b_j$, wherein $W = (\omega_1, \omega_2, \ldots, \omega_n)^T$ is weighted vector which is associated with F and $\omega_j \in [0, 1]$, $\Sigma\omega_j = 1$, and b_j is the jth big value in a group of data $(a_1, a_2, \ldots, a_n)$, the F function is called the n-dimensional ordered weighted averaging operator.

Assume that $a = [a^L, a^U] = \{x \mid a^L \leq x \leq a^U\}$; then a is an interval value. The F function has the following formula:

$$f_k\left(a^L, a^U\right) = \frac{a^U + r a^L}{r + 1}. \tag{6}$$

We transform the interval value to the value type through the following formula:

$$\mathrm{UR}_k(a) = f_k\left(a^L, a^U\right) = \begin{cases} a^U, & r \longrightarrow 0, \\ \dfrac{a^L + a^U}{2}, & r \longrightarrow 1, \\ a^L, & r \longrightarrow \infty, \end{cases} \tag{7}$$

where k is the number of user request indexes. $\mathrm{UR}_k(a)$ represents the normality request of the kth user. After normalizing every user requirement index, we can get the user expected value set UR.

3.3. Trust Degree Model

Definition 3 (trust degree). The trust degree of composite semantic service can be expressed as follows:

$$\mathrm{TD} = \langle \mathrm{TD}_F, \mathrm{TD}_\mathrm{Q} \rangle, \tag{8}$$

where TD_F is the trust degree of functional attributes and TD_Q is the trust degree of QoS.

Moreover the genic QoS parameters can be classified into two categories. One is recordable type. The execution value of this QoS type can be recorded in execution log at run time, such as response time and successful execution rate. The other one is unrecordable type. The kind of these QoS parameters cannot be recorded in execution log. It is only evaluated by user experience, such as cost and availability:

$$\begin{aligned} &\mathrm{TD}_\mathrm{Q} = \langle \mathrm{TD}_\mathrm{QR}, \mathrm{TD}_\mathrm{QU} \rangle, \\ &\mathrm{TD}_\mathrm{QR} = \langle \mathrm{TD}_\mathrm{time}, \mathrm{TD}_\mathrm{rate} \rangle, \ \mathrm{TD}_\mathrm{QU} = \langle \mathrm{TD}_\mathrm{cost}, \mathrm{TD}_\mathrm{available} \rangle. \end{aligned} \tag{9}$$

3.4. User Experience Evaluation Model. User experience represents the subjective feelings of past users. It can be evaluated by different parameters. According to [14], user experience is evaluated by click rate for getting the web service ranking with PageRank algorithm. In [15], they use usage frequency to evaluate user experience. Either click rate or usage frequency can only reflect overall impression of web service. It is ambiguity. Therefore in the paper we established a new user experience evaluation model according to user requirements. The new model consists of the local user ratings and the global user ratings. The global user ratings present the overall impression of web service. And the local user ratings are good complement to the global user ratings, which evaluate web service from several aspects. However the evaluated index of web service is basically provided by service provider or third parties according to their own professional knowledge. In fact the consumer cannot pay attention to all indicators, or the professional degree of consumer is not enough to give an accurate evaluation. So we use fuzzy logic to represent the user ratings. Fuzzy logic is based on fuzzy sets that represent vague data with the help of the so-called membership functions that represent the degree, referred to as membership, at which a certain datum belongs to a fuzzy data set.

Definition 4 (user experience evaluation model). The fuzzy representation is based on the assumption that the user ratings can be expressed as a number in the range $[0, 1]$. That means a user experience evaluation of web service can be presented by assigning values in the range $[0, 1]$. Thus user experience evaluation of web service is represented as a fuzzy set UE:

$$\mathrm{UE} = \left\langle \mathrm{UE}_g, \mathrm{UE}_l \right\rangle, \quad \mathrm{UE}_l = \{x, \mu_k(x) \mid x \in \mathrm{QoS}\}, \tag{10}$$

where UE_g is the global user ratings evaluation of web service. UE_l is the local user ratings evaluation according to QoS attributes. $\mu_k(x)$ represents the grade of membership of x evaluation index from consumer k. The membership function is a function of ratings. We define the membership function for x in a fuzzy set defining WS as follows:

$$\mu_k(x) = \begin{cases} 0, & 0 \le x \le \alpha, \\ \dfrac{1}{1 + [\beta / (x - \alpha)]}, & \alpha < x \le 100. \end{cases} \tag{11}$$

4. A Novel Composite Semantic Web Service Selection Approach

In this section, a novel composite semantic web service selection approach is proposed which takes the candidates' trust degree into consideration. Our selection approach mainly consists of the following five parts. In Part 1, we analyze execution log to get the real value of recordable QoS attributes. Compared to QoS value which is provided by service provider, calculate the trust degree of recordable QoS attributes. In Part 2, we compare the QoS constraint of user request with the recordable QoS attribute value to calculate the pass user satisfaction degree and compare pass user satisfaction degree with user experience evaluation for getting the credibility of pass user evaluation. In Part 3, we use credible local user evaluation to calculate the trust degree of the unrecordable QoS attributes. In Part 4, we use credible global user evaluation to calculate the trust degree of functional property. In Part 5, according to user requirement model, we consider the credible similar degree of IOPE and the credible QoS attribute parameters to select the best composite web service.

4.1. Analyze Execution Log. Traditionally, web services are individually deployed on proprietary infrastructure owed by the organization which operates and utilizes these services. With the increasing adoptions of "Platform as a service" paradigm, which provides a centralized runtime execution environment, more and more web services are published on centralized runtime execution environments, such as IBM Web Sphere Process Server, Microsoft Azure Services Platform, and Google App Engine. Adoptions of such a mode facilitate the monitoring of the services execution to obtain the execution logs.

Once a composite web service is deployed in a runtime execution environment, the composite web service can be executed in many execution instances of service composition. Each execution instance of service composition is uniquely identified with an identifier (id). In each execution instance, events can be triggered. We record the triggered events in the log using the logging facility provided by the execution environment. An execution log contains different types of events. For example, service error events are triggered when service error occurs. Service invocation events indicate the timeline of a web service execution.

We use service invocation events and service error events to evaluate the real value of the successful execution rate. $\mathrm{QR}_\mathrm{rate}$ is defined as the recorded successful execution rate attribute that can be calculated as follows:

$$\mathrm{QR}_\mathrm{rate} = 1 - \frac{N_\mathrm{error}}{N_\mathrm{invocation}}, \tag{12}$$

where N_{error} is the number of the service error events. $N_{\text{invocation}}$ is the total number of the service invocation events.

In particular, an ENTRY event is triggered when a service is invoked. An EXIT event occurs when a service completes the computation and returns results. Each event is recorded with the time of triggering, the name of the service which triggers the event, and the id of the execution instance and the underlying application. So we can get the response time from the ENTRY event to the EXIT event:

$$\text{QR}_{\text{time}} = \frac{1}{K}\sum_{k=1}^{K}\left(T_{\text{exit}} - T_{\text{entry}}\right), \tag{13}$$

where K represents the number of execution results and QR_{time} represents the recorded response time attribute. T_{exit} represents the triggering time of the EXIT event. T_{entry} is the triggering time of the ENTRY event.

For recordable QoS attributes, the distance between execution results with QoS information describes its trust degree. So the greater distance means the worse credibility. We calculate the distance to use the following formula:

$$\text{TD}_{\text{QR}}(i) = \begin{cases} 1 - \dfrac{\left|\text{QR}_i - \text{Q}_i\right|}{\text{Q}_i}, & \text{QR}_i < 2\text{Q}_i, \\ 0, & \text{QR}_i > 2\text{Q}_i, \end{cases} \tag{14}$$

where i represent the number of recordable QoS dimensions, QR_i represent the value of Q_i in execution log, and if $\text{QR}_i > 2\text{Q}_i$, that means the distance between execution results with QoS information is so big that the credibility of the service provider is 0.

4.2. User Satisfaction Degree. Then we use the gray correlation analysis method to get the user satisfaction degree. The gray correlation analysis method can obtain the relationship of two groups of sequences through calculating their distance [16]. So we can get the following formula:

$$\begin{aligned} d_k(i) &= \left|\text{UR}_i - \text{Q}_i\right|, \\ \text{US}_k(i) &= r\left(\text{UR}_i, \text{Q}_i\right) = \frac{\rho d_{\max}}{d_k(i) + \rho d_{\max}}, \quad \rho \in [0,1], \end{aligned} \tag{15}$$

where $\text{US}_k(i)$ represents the user satisfaction degree of the ith recordable user requirement index. $r(\text{UR}_i, \text{Q}_i)$ represents the correlation value between the user expected value and real value of operation. ρ represents the resolution value. $d_{\max}$ represents the max value of the distance of the user expected value and real value of operation.

Finally, we compare the user evaluation with user satisfaction degree. The distance of two values is closer; the trust degree of the user evaluation is higher. Assumed TD_k represents the trust degree of the kth user evaluation. We can get the following formula:

$$\begin{aligned} \text{TD}_k(i) &= 1 - \frac{\left|\text{US}_k(i) - \mu_k(i)\right|}{\mu_k(i)}, \\ \text{TD}_k &= \frac{\sum_{i=1}^{I}\text{TD}_k(i)}{I}, \end{aligned} \tag{16}$$

where I is the total number of the recordable QoS attributes.

4.3. Trust Degree of Unrecordable QoS Attribute. In this section we will use the user experience to calculate the credibility of unrecordable QoS dimensions. We use the user requirement to evaluate the bygone score of unrecordable QoS dimensions. If this user gives the superior limit, the bygone score should be computed using the following formula:

$$\text{BS}_k(j) = \begin{cases} \dfrac{\text{UR}_k(j) - \text{Q}_j}{\text{UR}_k(j) - \text{Q}_L} + 0.6, & \text{UR}_k(j) > \text{Q}_j, \\ 0, & \text{UR}_k(j) \le \text{Q}_j, \end{cases} \quad (1 \le j \le J). \tag{17}$$

Q_L represent the minimum value of the QoS dimension in formula (17). If users give the lower limit, the bygone score of unrecordable QoS dimensions should be computed using the following formula:

$$\text{BS}_k(j) = \begin{cases} \dfrac{\text{Q}_j - \text{UR}_k(j)}{\text{Q}_m - \text{UR}_k(j)} + 0.6, & \text{UR}_k(j) < \text{Q}_j, \\ 0, & \text{UR}_k(j) \ge \text{Q}_j, \end{cases} \quad (1 \le j \le J). \tag{18}$$

Q_m represent the maximum value of the QoS dimension in formula (18). J represents the number of unrecordable QoS dimensions, and k represents the number of users. We can calculate the distance of the bygone score and the credibility user comment to get the credibility of the unrecordable QoS dimension as the following formula:

$$\text{TD}_{\text{QU}}(j) = \frac{1}{K}\sum_{k=1}^{K}\left(1 - \frac{\left|\left(\text{TD}_k(j) \times \mu_k(j)\right) - \text{BS}_k(j)\right|}{\text{BS}_k(j)}\right), \quad (1 \le j \le J). \tag{19}$$

4.4. Trust Degree of IOPE. IOPE is the functional property of web service. So the trust degree of IOPE is due to the global user evaluation. The following formula helps us to get TD_F:

$$\begin{aligned} \text{TD}_F &= \frac{1}{K}\sum_{k=1}^{K}\left(\text{TD}_k \times \text{UE}_g\right) \\ &= \frac{1}{K}\sum_{k=1}^{K}\left(\frac{\sum_{j=1}^{J}\text{TD}_k(j)}{J} \times \text{UE}_g\right), \end{aligned} \tag{20}$$

where K is the number of consumers. J is the number of the unrecordable QoS attributes. UE_g represents the global user experience evaluation.

4.5. The Novel Web Service Selection Approach. As mentioned above, we compute the trust degree of the recordable QoS attributes and the unrecordable QoS attributes, respectively. Finally we can use formula (21) to get the evaluation result of the composite semantic web service. Among all the candidate

Table 1: User requirement indexes.

Indexes	Cost	Response time	Successful execution rate	Availability
Type	Unrecordable	Recordable	Recordable	Unrecordable
Constraint (weight)	8 (0.2)	0.3 s (0.2)	>80 (0.3)	>95 (0.3)

services, the service with the highest score of evaluation is selected:

$$\begin{aligned} \mathrm{WS} &= \mathrm{WS}_F + \mathrm{WS}_{\mathrm{Q}} \\ &= W_F \times \mathrm{TD}_F \times \mathrm{Sim}_F + \sum_{i=1}^{I} \left(W_{\mathrm{Q}i} \times \mathrm{TD}_{\mathrm{QR}}(i) \times \mathrm{Q}_i\right) \\ &\quad + \sum_{j=1}^{J} \left(W_{\mathrm{Q}j} \times \mathrm{TD}_{\mathrm{QU}}(j) \times \mathrm{Q}_j\right). \end{aligned} \tag{21}$$

It is supposed that I recordable QoS dimensions and J unrecordable QoS dimensions are considered, and each candidate service is executed K times. That means we have K pieces of execution results and user experience evaluations. The credibility of recordable QoS dimensions is computed separately and each piece of execution log is used once, so the time cost is $O(I \times K)$. During computing the credibility of unrecordable QoS dimensions, the time cost is $O(N \times M)$. M represents the number of web service composition nodes and N represents the number of candidate web services in every node. The complexity of the proposed algorithm which calculates the evaluation of all candidate services is $O(N \times M \times (I \times K + J))$.

5. Case Study

In this section we design two experiments to evaluate the performance of the proposed composite web service selection method. The experiments have been performed on a PC powered by an AMD Quad Core A4, 1.5 GHZ processor, equipped with 4 GB RAM, and a 500 GB hard disk, and the software environment of the experiments is Win 8 SP1, Java 1.6. Our objective is to prove the availability of our proposed composite service selection method. For this purpose, we adopt the traditional web service selection based on QoS and IOPE evaluation to compare with our approach. It does not consider trust degree of QoS and IOPE information in traditional composite web service selection method. According to QoS value and IOPE similar degree, it uses formula (22) to sort the candidate web service. Consider

$$\mathrm{WS} = \mathrm{WS}_F + \mathrm{WS}_{\mathrm{Q}} = W_F \times \mathrm{Sim}_F + \sum_{i=1}^{I} \left(W_{\mathrm{Q}i} \times \mathrm{Q}_i\right). \tag{22}$$

The test case is a web service composition that implements a travel planning process. It looks for tourist destination, books flight ticket and hotel reservation in parallel, and finally invokes a car rental operation. Per each of the tasks in the process, there are 10 candidate services, distributed among the servers that fulfill the required functionality and offer different QoS. Firstly we give their requirement indexes which are presented in Table 1.

Table 2: The percentage of unauthentic candidate web service.

Tasks	Case 1	Case 2	Case 3	Case 4
Looking destination	20%	40%	60%	80%
Booking ticket	30%	50%	70%	80%
Hotel reservation	20%	40%	60%	80%
Car rental operation	10%	30%	50%	80%

Table 3: The first experiment result.

	Case 1	Case 2	Case 3	Case 4
The selected probability				
Traditional	0.14	0.35	0.52	0.76
Proposed	0.10	0.18	0.24	0.48
The fitness value				
Traditional	0.91	0.78	0.65	0.40
Proposed	1	1	1	0.8

Two experiments are designed to illustrate the availability of the novel proposed approach. Every method will execute 50 times. For the first experiment, the traditional algorithm and the proposed algorithm run under four different cases to monitor the influence of two methods as the number of the unauthentic considered services increases. Table 2 shows the different proportion of unauthentic candidate web services. The first experiment results are given in Table 3. For the second experiment, the traditional algorithm and the proposed algorithm run with four different groups of exaggerated degree of QoS and IOPE under Case 2. The second experiment results are given in Table 5.

Table 3 shows the selected probability of unauthentic candidate web services and the fitness value. The selected probability of unauthentic candidate web services can be calculated by the following formula:

$$P = \frac{N_S}{4 \times 50}, \tag{23}$$

where N_S is the number of selected unauthentic web services. It can be seen from Table 3 that the selected probability of proposed approach is less than the traditional approach under all the four cases, which illustrates that the proposed approach can filter unauthentic web service effectively. It can also be seen that the selected probability of proposed approach did not rise as the number of the unauthentic considered services increases, which illustrates that the proposed approach is not influenced by the number of unauthentic web services. Furthermore, Table 3 also shows that the fitness values of proposed approach under all the four cases are equal to one, which means that the selected services set of proposed approach can satisfy user's constraint condition under each

Table 4: The exaggerated degree of QoS and IOPE.

Indexes	Cost	Response time	Successful execution rate	Availability	IOPE
Group 1	0.1	0.2	0.15	0.2	0.2
Group 2	0.3	0.4	0.5	0.5	0.5
Group 3	0.6	0.7	0.7	0.6	0.6
Group 4	0.8	0.8	0.8	0.8	0.8

Table 5: The second experiment result.

	Group 1	Group 2	Group 3	Group 4
The selected probability				
Traditional	0.55	0.68	0.80	0.85
Proposed	0.18	0.14	0.06	0.03
The fitness value				
Traditional	0.78	0.64	0.42	0.20
Proposed	1	1	1	1

case. But the traditional approach cannot fully satisfy the user's needs under the influence of unauthentic candidate web services.

In the next step we assume that the probability of unauthentic web service is fixed under Case 2. We increase the exaggerated degree of QoS and IOPE information to monitor the influence of two methods. As mentioned above in QoS model, the QoS values are four-dimensional: cost, response time, successful execution rate, and availability. There services have been registered into service database. They executed several times. The database establishes the execution logs to record historical data and collect the user ratings to evaluate the user experience. Table 4 shows the exaggerated degree.

Table 5 shows the second experiment results. It can be seen from Table 5 that the exaggerated degree of QoS and IOPE is higher, the selected probability of proposed approach is lower, but the selected probability of traditional approach is higher, which illustrates that the proposed approach cannot influence by the exaggerated degree of QoS and IOPE. It can also be seen that the fitness values of proposed approach under different exaggerated degree are equal to one, which means that the selected services set of proposed approach can satisfy user's constraint condition under different exaggerated degree. But the fitness value of traditional approach is lower when the exaggerated degrees increase, which illustrates that the traditional approach seems to opt for more unauthentic candidate web services.

6. Conclusion

In this paper, the content of the research is to propose a novel trust-aware composite semantic web service selection approach. In order to filter exaggerated QoS and IOPE information, this paper established a trust degree model. According to the execution log and user experience, we calculate the credibility of QoS information and IOPE similar degree. Then we get the best candidate web service based on trustworthy QoS and IOPE. Finally, through two experiments we proved that the new method can effectively avoid the influence of web services which include exaggerated and unauthentic service profile.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

Denghui Wang would like to extend sincere gratitude to corresponding author, Hao Huang, for his instructive advice and useful suggestions on this research. And the authors thank the anonymous reviewers for their valuable feedback and suggestions.

References

[1] A. Furno and E. Zimeo, "Context-aware composition of semantic web services," *Mobile Networks & Applications*, vol. 19, no. 2, pp. 235–248, 2014.

[2] H. Zheng, W. Zhao, J. Yang, and A. Bouguettaya, "QoS analysis for web service compositions with complex structures," *IEEE Transactions on Services Computing*, vol. 6, no. 3, pp. 373–386, 2013.

[3] S. Wang, X. Zhu, and F. Yang, "Efficient QoS management for QoS-aware web service composition," *International Journal of Web and Grid Services*, vol. 10, no. 1, pp. 1–23, 2014.

[4] C.-F. Lin, R.-K. Sheu, Y.-S. Chang, and S.-M. Yuan, "A relaxable service selection algorithm for QoS-based web service composition," *Information and Software Technology*, vol. 53, no. 12, pp. 1370–1381, 2011.

[5] Z. Yanwei, N. Hong, D. Haojiang, and L. Lei, "A dynamic web services selection based on decomposition of global QoS constraints," in *Proceedings of the IEEE Youth Conference on Information, Computing and Telecommunications (YC-ICT '10)*, pp. 77–80, November 2010.

[6] Z.-Z. Liu, X. Xue, J.-Q. Shen, and W.-R. Li, "Web service dynamic composition based on decomposition of global QoS constraints," *International Journal of Advanced Manufacturing Technology*, vol. 69, no. 9–12, pp. 2247–2260, 2013.

[7] D. Huijun, Q. Hua, Z. Jihong, D. Wenhan, and X. Wujie, "A distributed optimal scheme based on local QoS for web service composition," *China Communications*, vol. 11, no. 13, pp. 142–147, 2014.

[8] J. A. Parejo, S. Segura, P. Fernandez, and A. Ruiz-Cortés, "QoS-aware web services composition using GRASP with Path Relinking," *Expert Systems with Applications*, vol. 41, no. 9, pp. 4211–4223, 2014.

[9] A. Missaoui, "A QoS-based neuro-fuzzy model for ranking web services," in *Proceedings of the 3rd International Conference on Information Technology and e-Services (ICITeS '13)*, pp. 1–5, March 2013.

[10] B. Pernici and S. H. Siadat, "Evaluating web service QoS: a neural fuzzy approach," in *Proceedings of the IEEE International Conference on Service-Oriented Computing and Applications (SOCA '11)*, December 2011.

[11] T. Zhang, J. Ma, Q. Li, N. Xi, and C. Sun, "Trust-based service composition in multi-domain environments under time constraint," *Science China Information Sciences*, vol. 57, no. 9, pp. 1–16, 2014.

[12] R. Zhu, H.-M. Wang, and D.-W. Feng, "Trustworthy services selection based on preference recommendation," *Journal of Software*, vol. 22, no. 5, pp. 852–864, 2011.

[13] W. Denghui, H. Hao, and X. Changsheng, "A novel web service composition recommendation approach based on reliable QoS," in *Proceedings of the IEEE 8th International Conference on Networking, Architecture and Storage (NAS '13)*, pp. 321–325, IEEE, Xi'an, China, July 2013.

[14] C. Li, B. Cheng, J. Chen, P. Gu, N. Deng, and D. Li, "A web service performance evaluation approach based on users experience," in *Proceedings of the IEEE 9th International Conference on Web Services (ICWS '11)*, pp. 734–735, July 2011.

[15] X. Huang, "UsageQoS: Estimating the QoS of web services through online user communities," *ACM Transactions on the Web*, vol. 8, no. 1, article 1, 2013.

[16] J. Deng, *Gray Control System*, Huazhong Institute of Technology Press, Wuhan, China, 1985.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 747392, 9 pages
http://dx.doi.org/10.1155/2015/747392

Research Article

A Revenue Maximization Approach for Provisioning Services in Clouds

Li Pan[1] and Datao Wang[2]

[1]*School of Computer Science and Technology, Shandong University, Jinan 250101, China*
[2]*Jinan Resident Office of CNAO, Jinan 250011, China*

Correspondence should be addressed to Li Pan; panli@sdu.edu.cn

Received 27 March 2015; Accepted 16 June 2015

Academic Editor: Bao Rong Chang

With the increased reliability, security, and reduced cost of cloud services, more and more users are attracted to having their jobs and applications outsourced into IAAS data centers. For a cloud provider, deciding how to provision services to clients is far from trivial. The objective of this decision is maximizing the provider's revenue, while fulfilling its IAAS resource constraints. The above problem is defined as IAAS cloud provider revenue maximization (ICPRM) problem in this paper. We formulate a service provision approach to help a cloud provider to determine which combination of clients to admit and in what Quality-of-Service (QoS) levels and to maximize provider's revenue given its available resources. We show that the overall problem is a nondeterministic polynomial- (NP-) hard one and develop metaheuristic solutions based on the genetic algorithm to achieve revenue maximization. The experimental simulations and numerical results show that the proposed approach is both effective and efficient in solving ICPRM problems.

1. Introduction

Cloud computing is holding a promising approach to deliver on-demand computing services to a wide range of consumers in a pay-as-you-go way in these years. With virtualization as a key enabler, cloud computing delivers Infrastructure As A Service (IAAS) that integrates computation, storage, and networking resources in a virtualized environment [1]. Virtualization technology makes the independence of applications and servers feasible through consolidating multiple applications and jobs running in different virtual machines (VMs) on a single physical machine (PM).

Virtualization not only brings cloud providers efficiency gains in terms of processing power but also saves electric power, space, and cooling, since the number of physical machines running is greatly reduced [2]. Recently, there is a significant increase in both the supply and demand sides of this new market for IAAS cloud services. It represents a new business model where clients buy job execution services from clouds while the cloud providers gain revenues. Such IAAS environments offer elastic job execution services with flexible QoS in a way that they can be measured, thus allowing for a pricing mechanism to be enforced. Clients often have different preferences on service qualities and each client is characterized by a willingness-to-pay function. This function is usually parameterized with both the clients own payment willingness and QoS of the offered service. Based on payments from clients, cloud providers try to maximize their financial revenue through provisioning services.

The above problem is defined as IAAS cloud provider revenue maximization (ICPRM) problem in this paper. We address this problem through provisioning services of flexible QoS levels to clients. Briefly speaking, we equip the provider with an admission control mechanism that helps the provider to determine which subset of clients to admit and what QoS levels to offer to each admitted client, in order to maximize his revenue. We formulate the revenue maximization problem here as a relaxed Multiple-Choice Knapsack Problem (MCKP), which is a variant of the classical 0/1 Knapsack Problem (KP). Since to find an exact solution for a MCKP is an NP-hard problem, in this paper we propose a genetic algorithm to obtain a feasible near-optimal solution. And we evaluate our revenue maximization mechanisms for provisioning services in a simulated environment to illustrate the efficiency and effectiveness of the proposed approach.

The remainder of the paper is organized as follows. Section 2 gives an overview of the IAAS cloud computing environment. The optimization problem and the proposed genetic optimization algorithm are presented in Sections 3 and 4. Section 5 outlines simulation experimental results. Related work is discussed in Section 6 while concluding remarks are found in Section 7.

2. System Model

This section presents an overview of the cloud platform environment and the client model assumed in our work, which is based on an IAAS model. For clarity, in this paper we focus on a simple type of IAAS service: provisioning virtual machine in a provider's cloud datacenters to run jobs, especially batch type jobs submitted by clients.

2.1. IAAS Cloud Platform Description. We consider a scenario where a cloud provider hosts multiple clients' jobs in a shared IAAS platform. Clients who have jobs to run gain access to the cloud resources by requesting services from the provider. The service proposed by the cloud provider in our approach is offering infrastructures to the clients in order to run their jobs. Whether jobs are run in virtual machines or raw physical machines, they are transparent to clients and they only care for the quality of the service the cloud provider offered to run their jobs. And the role of this work is to help the provider to maximize its revenue while proposing and provisioning services.

A high-level structural view of the IAAS environment considered in this work is depicted in Figure 1. As shown in the figure, an IAAS provider's physical infrastructure consists of a number of computing servers, which are equipped with physical resources such as CPU, memory, and I/O bandwidth. On top of the physical infrastructure is the virtualization layer created by isolated VMs. Virtualization mechanisms partition the physical resources into multiple isolated virtual machines. Specifically, in this paper for clarity we only consider the number of CPUs allocated to a virtual machine for provisioning services to run jobs, as the processing capacity measurement criteria. Thus a VM's processing performance is characterized by its allocated virtual CPU cores. The different numbers of CPUs allocated to a client's VM would result in different service qualities for running jobs. It is the responsibility of the *service provision manager* to admit jobs submitted by clients and determine how many CPUs are required to provision virtual machines for running the submitted jobs.

The above-described hosting environment in fact forms a cloud service market where a cloud provider sells IAAS services rather than raw machine resources and clients make payment as returns to the provider for running their jobs. Each client desires a performance bound, that is, QoS, for its job execution service, and pays corresponding prices. In this work we consider a bidding approach, upon which each client submits a bid to the cloud provider to obtain job execution services. Each client has a utility function that gives its budget value as a function of a range of service quality levels. We will discuss the utility functions in the following subsection. The cloud provider selects clients for provisioning capacities based on clients' bids for submitted job execution service requests. In the cloud provider's side, a service provision manager is responsible for admitting clients to run their jobs. When faced with a bunch of clients, the admission control includes deciding which clients to admit, in what price to provision the services, and in what QoS levels. We believe that for a batch type job execution cloud service one performance measure that matters most to the clients is *job response time*, that is, expected job completion time [2]. Thus in this paper we will take the job response time as the criteria of service QoS levels.

Given the framework described, we define the provider's business objective, formulated as ICPRM problem, as the optimal provisioning of capacities to VMs to host clients' jobs so as to maximize his overall revenue.

2.2. Utility Functions. In this subsection we discuss the utility functions, based on which the clients specify bids and the provider makes admission control and virtual machine provision decisions.

In this work we assume that a client has a flexible but constrained requirement on the response time of his jobs. This means that a client has a rigid deadline requirement before which the job must be finished; otherwise he would not pay for the job execution service. On the other side, the client would pay more for a job finished earlier than its rigid deadline. For example, a client may have 20 hours as the deadline for a job he submits to a cloud platform, but he would pay more if the job will be finished sooner. Thus, for each client, a utility function specifies the value he is willing to pay to the provider for a range of service quality levels—expected job completion time acceptable to the client. This utility function is also called the client's willingness-to-pay function.

The formulation of utility functions must be simple, rich, and tractable. We assume that a client has a basic utility gain (or revenue) from finishing the job and a convexly increasing utility loss for the delay in service response time t_i. And we model utility function $u_i(t_i)$ of a client for a job execution service provided as

$$u_i(t_i) = a_i + b_i * \left(1 - \left(\frac{t_i - t_{i,\min}}{t_{i,\max} - t_{i,\min}}\right)^{\alpha_i}\right), \quad (1)$$

where a_i, b_i, α_i, $t_{i,\min}$, and $t_{i,\max}$ are user-defined variables, with a_i denoting the base payment for running the job, b_i denoting the floating return for accomplishing the job earlier than the deadline, $t_{i,\min}$ denoting the earliest job finish time reasonable, and $t_{i,\max}$ denoting the maximum completion time acceptable by the client i, beyond which he would not make payment for the service. And α_i is a client specific parameter characterizing how he is sensitive to service response time delay. A set of example utility functions is shown in Figure 2. By modeling in this way, a client's utility functions u_i are concave and monotonically decreasing in t_i, which is reasonable in batch type job execution domains.

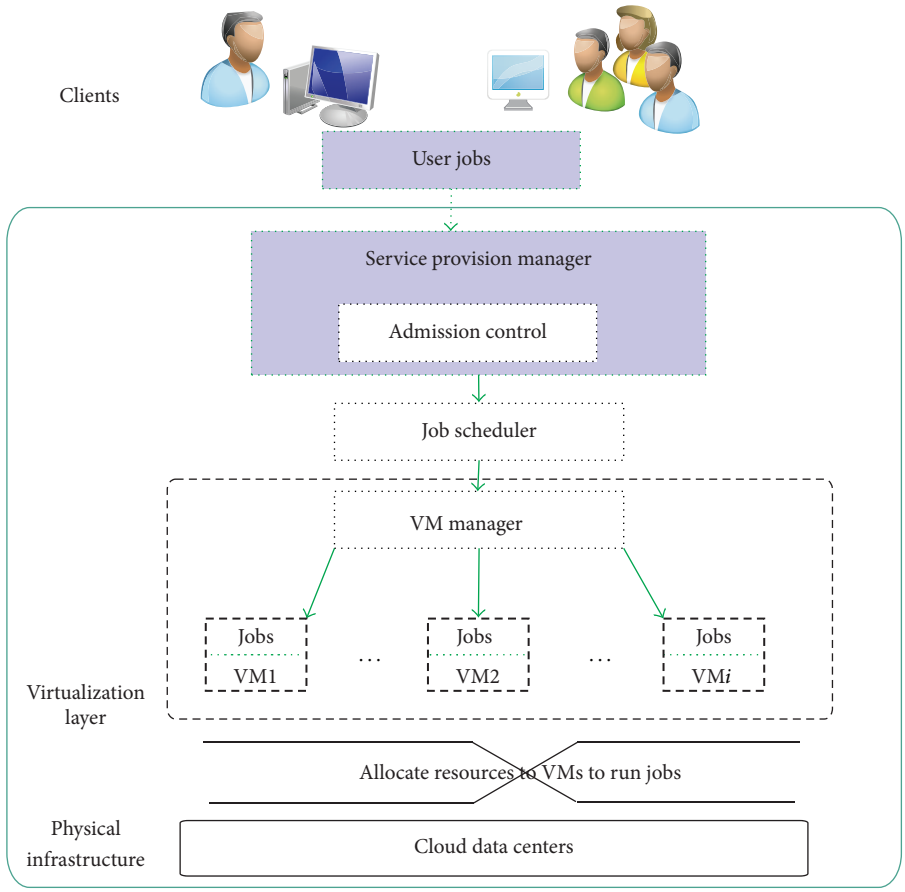


Figure 1: System model.

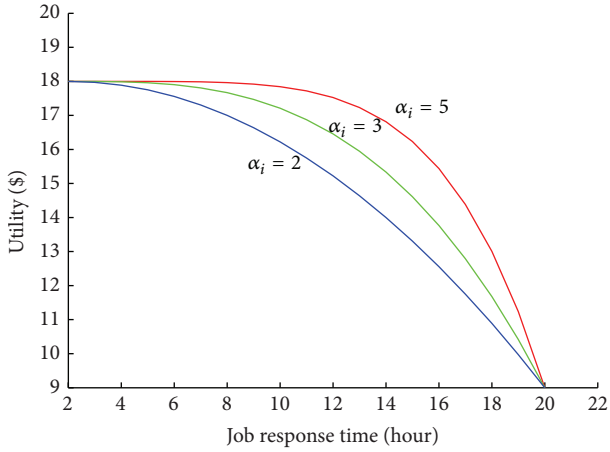


Figure 2: Example utility functions. α_i is set by a client and denotes how he is sensitive to service response time delay.

2.3. How Much to Provision: Modeling Batch Type Jobs Execution in Clouds. The goal of our provision mechanism is for the IAAS provider to optimally decide the QoS levels and allocate specific number of CPU cores to provision virtual machine to run clients' jobs. A significant problem for the provider to provision CPUs to run a job is to decide how many CPU capacities are needed to meet a specific QoS demand, that is, job response time in this paper. This is a reverse problem of determining the completion time of a job given the physical capacity of the machine on which it runs. Estimating job completion time given specific processing capacity is a complex problem that has been extensively researched and it falls beyond the scope of this paper. We acknowledge that it is sometimes difficult to estimate accurate execution time given some types of jobs. And whether our revenue maximization driven virtual machine provisioning approach is effective or not is highly dependent on the accuracy of the estimation of job execution time. However, for batch type jobs we discussed in this paper, we argue that it is possible to build fairly accurate models. This model can be built by benchmark methods. And in this paper, for clarity reasons, we assume that job execution time T has a certain relationship with the number of CPUs allocated to the VM it runs, represented as n, as follows:

$$T(n) = \frac{t}{\ln(n) + 1}, \tag{2}$$

where t corresponds to the time it would take to complete the job if only one general CPU core is provisioned to a VM to run the job. With this equation we model the fact that job execution time will reach a saturation point and will improve only marginally after reaching a certain number of CPUs.

3. Revenue Maximization Problem Formulation

3.1. Problem Formulation. We model our IAAS service platform to have a certain number of available identical CPU cores, denoted as C, which are running in a cluster of machines maintained by the cloud provider. Consider a given arbitrary set of clients I such that the ith client ($i \in I$) has a job to execute, the acceptable response time range $[t_{i,\min}, t_{i,\max}]$, and the utility function u_i for specifying payment based on the guaranteed service response time t_i, that is, job completion time. It is assumed that the variable values of response time are not taken continuously from the acceptable range of a client but discretely. This assumption is reasonable due to the fact that end users are usually not sensitive to response time difference of a too small time scale; that is, end users may feel no difference between spending, for example, 1 minute and spending 2 minutes to get a job finished [2]. For example we may consider a granularity of 30 minutes.

As described above, our objective is to maximize the provider's revenue within its CPU capacity constraints, and the mathematical formulation of the ICPRM is as follows:

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in I} x_i u_i\left(t_i\right), \\ \text{subject to} \quad & t_i \in T_i = \left\{\tau_{i,1}, \ldots, \tau_{i,e}\right\} \subset \left[t_{i,\min}, t_{i,\max}\right], \\ & \qquad\qquad i \in I, \quad (3) \\ & x_i \in \{0,1\}, \quad i \in I, \\ & \sum_{i \in I} T^{-}\left(t_i\right) \le C, \quad i \in I, \end{aligned}$$

where $T^{-}(t_i)$ denotes CPU provision decision function for the ith client, x_i is a pseudo-Boolean integer decision variable to determine whether the ith client is admitted or not, and t_i is a variable to denote the response time for the ith client's job. The CPU provision decision function $T^{-}(t_i)$ is a reverse function of (2). The job response time set $T_i = \{\tau_{i,1}, \ldots, \tau_{i,e}\}$ is derived discretely from the ith client's acceptable range $[t_{i,\min}, t_{i,\max}]$. We may without loss of generality assume $\tau_{i,1} = t_{i,\min}$ and $\tau_{i,e} = t_{i,\max}$.

The above ICPRM problem is a relaxed Multiple-Choice Knapsack Problem (MCKP), where there are groups of items with the constraint that exactly an item can be picked from each group, and for ICPRM problem the restriction of picking exactly one item from each group is relaxed; that is, you can either pick one item from a group or leave it. Leaving a job unserviced models the situation that the provider denies a client either because servicing it is nonprofitable or because there are not enough CPU capacities left.

3.2. Optimization Technique. Now let us study the above profit maximization problem and discuss the optimization techniques for solving it.

Theorem 1. *The optimization problem ICPRM is NP-complete.*

The theorem's proof is a straightforward reduction from the 0-1 Knapsack Problem. Since we only consider a finite number of QoS level alternatives, it is tempting to do a complete enumeration to determine the optimal solution for a small problem size. However, with the increasing number of potential clients faced by the provider, finding exact solutions for this NP-complete problem would incur huge and usually unacceptable computation cost. Thus, we resort to heuristic algorithms for near-optimal solutions. There exist a number of heuristics in the literature for MCKP and KPs in general. For example, greedy approaches have been proposed to find near-optimal solutions of KPs [3, 4]. Even though greedy approaches are simple to implement and are widely used, the optimality cannot often be guaranteed when trapped in local optimum. Thus, in this work we propose a metaheuristic genetic algorithm (GA) instead of greedy heuristics to handle this computationally costly optimization problem.

4. Genetic Algorithm Optimization for ICPRM

In the previous section we have shown that the optimization problem ICPRM is NP-hard. We now develop a metaheuristic genetic algorithm in order to find a near-optimal solution to this revenue maximization problem.

4.1. Review of Genetic Algorithm (GA). Genetic algorithms (GA) have been considered as effective techniques for solving complex operational research problems. A genetic optimization algorithm can be considered as an iterative process to search the best solutions from a predefined sized population of chromosomes for a given problem. The search usually starts from an initial randomly generated population of individuals. During every evolution step, a fitness function is used to evaluate individuals, and then the operators of reproduction, crossover, and mutation are invoked to create offsprings. In these steps the individuals with high fitness value will have a higher probability to reproduce.

4.2. GA for ICPRM Problem. In this section, we describe the formulation of a genetic optimization algorithm for ICPRM problem. To cope with the ICPRM problem, we propose the GA features relevant to solution encoding, fitness value, elitist reproduction, and crossover operation during the evolution process.

4.2.1. Solution Encoding. When applying genetic optimization algorithms to solve real-life operational application problems, the hardest step is usually how to encode the solutions as chromosomes. The chromosomes are often represented by strings and thus genetic operators can be applied on them. In this work, we propose an encoding scheme for revenue maximization as shown in Figure 3.

A solution is represented by a string of length equal to the number of variables, which is the number of clients currently requesting services from the provider. Each position, i, of the string, which represents the ith gene in the chromosome, can take any integer from $\{0, \ldots, e_i\}$, where an integer "0" in position i represents the ith client that will not be admitted

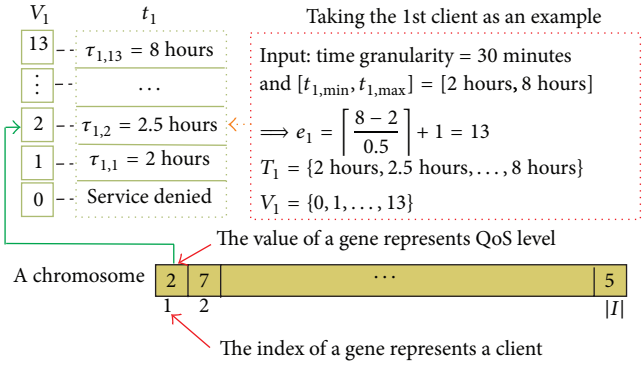


FIGURE 3: Solution encoding.

while one of the other integers in the same position, say j, represents the ith client that will be admitted and serviced with job execution response time $T_i[j]$, which is the jth element in vector T_i, that is, $\tau_{i,j}$. This is illustrated by the following example.

Example 2. Let $|I| = 5$ and $e_i = 5$ for all i. Then one string representing a chromosome is an element of $\times_{i=1}^{5}\{0, 1, 2, 3, 4, 5\}$. A particular instance $(5, 2, 4, 0, 1)$ represents the solution which admits the 1st client's job with completion time guaranteed at $t_1 = \tau_{1,5}$, the 2nd with $t_2 = \tau_{2,2}$, the 3rd with $t_3 = \tau_{3,4}$, and the 5th with $t_5 = \tau_{5,1}$, respectively, while denying the 4th client's service request.

4.2.2. Fitness and Constraints. The fitness function measures to what extent the value of an individual chromosome satisfies the objective. For an individual feasible solution, its fitness value is the overall revenue gained by admitting the set of clients with the QoS levels denoted in it. Besides fitness value, a feasible solution must meet CPU constraints in our ICPRM problem. This means that during the whole evolutionary process all the constraints should be met anytime. This requires the operations of individual chromosome initialization, crossover, and mutation in our approach be modified accordingly. Thus, while creating the initial population, if the required overall amount of CPU capacities of a randomly generated individual violates the constraints, it must be rejected and regenerated. Similarly, during the crossover and mutation process, if an offspring or mutated individual violates the CPU constraints, it should be also given up and generated again.

4.2.3. Elitist Reproduction. During the evolution process, our GA identifies itself with its elitist reproduction schemes, which we will discuss below. At the beginning of GA, the initial population P^0 is created by randomly selecting N_p members from the candidate set, where N_p is the size of the population. When creating mating pools MP, in our approach an *elitist reproduction* replaces the usual probabilistic reproduction, by firstly cloning N_r top solutions. N_r is usually fixed to 10% of the population size. The advantage of using an elitist reproduction is that it can help prevent the loss of good solutions once they are found and thus the best solution is monotonically improving from generation to generation. After elitist reproduction, *tournament selection* is then invoked, during which a group of k members are selected to create a tournament. Then, the highest fitness valued individual from the tournament is picked out and placed in the mating pool. This process continues until the mating pool reaches the predefined size.

4.2.4. Evolution Process. During every evolution process, we apply standard cloning, crossover, and mutation operations over the individuals in the mating pool to create offsprings. While doing cloning operation, an individual is directly copied from the mating pool to the offspring population. After cloning operations, we use crossover operations to exchange the randomly selected fields (QoS level indexes) of two randomly selected parents to obtain two offsprings. Concerning mutation, we randomly replace QoS level indexes encoded on a chromosome by other QoS level indexes from the permissible QoS domains of a client. And the operations of cloning, crossover, and mutation continue until the offspring population with size N is created. The exit criteria in our genetic algorithm can be defined by a maximum number of generations (evolutions) or the situation occurs when the best feasible solution has not been updated for a few generations.

The genetic algorithm described above is summarized by the pseudocode in Algorithm 1. Note that the best feasible solution is updated at each generation.

5. Performance Analysis and Discussion

In order to evaluate the effectiveness and the efficiency of the proposed genetic based revenue maximization algorithm for the ICPRM problem, a simulation framework is implemented. We have changed the size of submitted jobs and the number of alternative job response time levels to test the performance of the proposed approach. Besides, we compare our proposed genetic optimization based approach with other methods. The detailed simulation setups, baseline heuristics, and numerical results of this implementation are explained next.

5.1. Experimental Methodology. For simulations, experimental settings and model parameters are chosen based on true-to-life IAAS cloud environments. Since higher load factors (the total demanded capacity divided by total capacity) increase the importance of an optimal capacity provision approach, for each test case the number of total available capacities is randomly set in a way that the load factors vary between 1.1 and 1.5. The number of clients varies between 10 and 100. For simplicity of implementation and demonstration, we assume that each client has the same range of acceptable job response time which is from 2 hours to 8 hours. Thus, different time granularity would result in varied number of alternative job response time levels. For example, when taking 30 minutes as the granularity, the number of alternative response time levels would be 12; but when taking 1 hour as the granularity, the number of alternative response

LET
N_p be the size of the GA population;
P_c be the cross probability and P_m be the mutation probability;
N_r be the elitist reproduction percentage;
INITIALIZE
set best revenue value so far to zero;
create the initial population P^0 by randomly selecting N_p feasible individuals from $V_p = V_1 \times V_2 \cdots \times V_{|I|}$;
evaluate the fitness value of each individual in P^0 and sort them by decreasing order;
update best solution so far as the best individual in P^0 and set best revenue value correspondingly;
WHILE NOT(Reached the exit criteria)
  P^{i+1} = ElitistReproduction(P^i, N_r); //to include the first N_r solutions in P^{i+1};
  MP = TournamentSelection(P^i);
  DO UNTIL Size(P^{i+1}) = N_p
    Q_c = Crossover(MP, P_c);
    Q_m = Mutation(MP, P_m);
    $P^{i+1} = P^{i+1} \bigcup Q_c \bigcup Q_m$;
  END UNTIL
  UPDATE best solution and best revenue value so far;
END WHILE
OUTPUT best solution and best revenue value;

Algorithm 1: Genetic algorithm for ICPRM problem.

time levels would be 6. And in the following experiments, we change the granularity to get varying number of response time levels.

5.1.1. Heuristics for Comparison. In order to study the runtime performance of our genetic algorithm for revenue maximization, we implement it along with a greedy approach. The greedy approach is based on a linear search of all feasible QoS levels and picks the highest-valued QoS level for each client. The value of each QoS level is calculated as the ratio between the revenue gained from it and the number of CPUs needed. And then the set of clients are sorted based on their highest values of feasible QoS levels. Finally the subset of clients with their guaranteed job response time levels are selected by packing them into the limited CPU capacities one by one from the sorted set of clients (from the highest to the lowest), until no more clients can be admitted. In this paper, we call this heuristic Greedy-HEU.

We implement all the algorithms in C programming language. The parameters in genetic based algorithm are set as follows:

(i) The size of the population is 100.

(ii) The maximum generation is 100.

(iii) The cross probability is 0.7 and the mutation probability is 0.1.

To avoid biasing results due to randomness, our GA executions are repeated 10 times, and average values are used.

5.2. Efficiency. We take the average CPU execution time to test the efficiency of the proposed genetic optimization based approach for the revenue maximization problem. We consider that the size of the jobs submitted by clients varies from 10 to 100 with increment of 10, and the number of alternative job response time levels changed from 6 to 14, with increment of 2. We ran all the simulations on an Intel 1.70 GHz PC with 3.69 G of RAM under Windows.

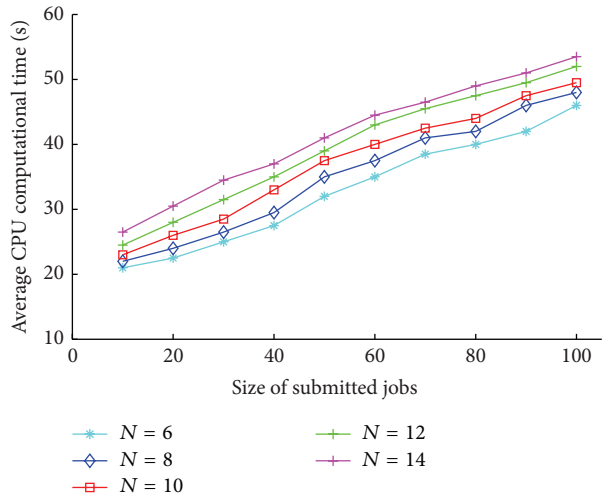


Figure 4: Average CPU computational time of our GA revenue maximization algorithm for different problem sizes. The parameter N is the size of the alternative response time levels.

We first test how our proposed genetic algorithm works for different problem instance sizes. The summary of results is shown in Figure 4. It can be seen from the figure that with the growth of job population size the computation time under the same number of alternative job response time levels increases. However, the computation time increases very slowly. Similarly, with the increment of alternative job execution time levels, the computation time under the same

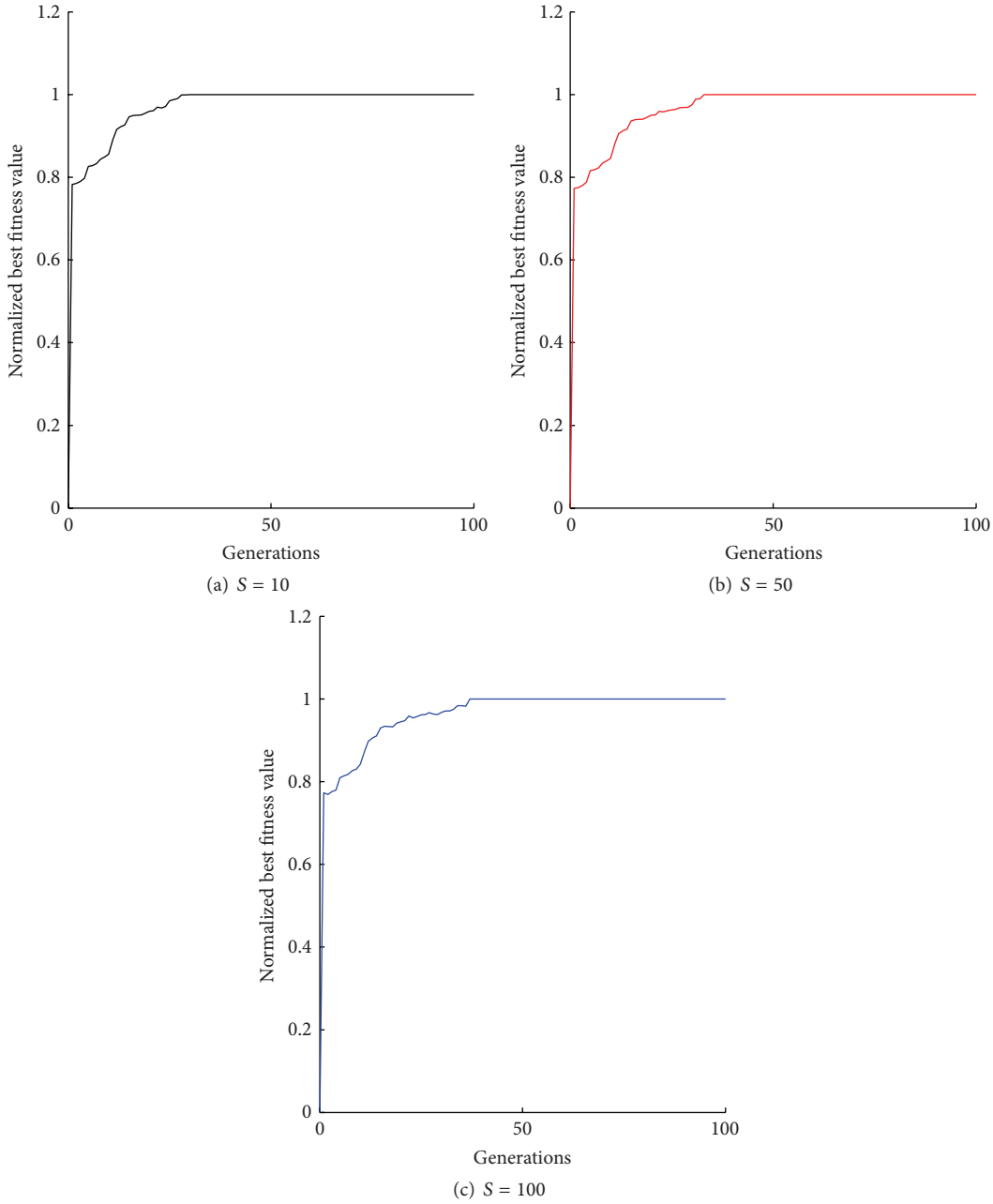
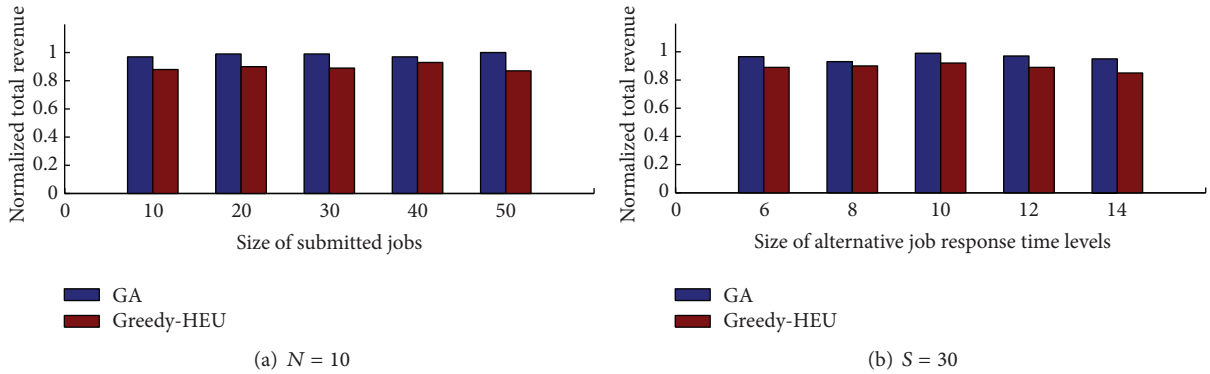


Figure 5: Examples of evolution trace. The parameter S is the size of the submitted jobs.

number of job population sizes also increases slowly. And for the biggest problem size tested in our simulation framework, the execution time of our proposed algorithm is no bigger than 60 seconds. This solving scale suits the requirements of most cloud provider revenue maximization problems.

Besides, the numbers of generations for our proposed genetic algorithm to obtain near-optimal solutions are less than 30 for the different problem instance sizes tested. Figure 5 shows three examples of the execution trace of our GA for solving ICPRM problems. The three examples are all parameterized as the alternative response time levels are 10 but with different job number sizes, as shown in the figure. As can be seen, it converges to the near-optimal solution, after about 30–40 generations in these examples. Overall, the proposed GA is efficient since near-optimal solutions for problems of reasonable sizes are obtained in at most 60 seconds and hence the approach is suitable for online implementations.

5.3. Effectiveness. We also conduct experiments to test the effectiveness of the proposed GA based optimization approach to find the global optimal solution for ICPRM problem. Figure 6 reports the optimal result obtained by our proposed GA optimization algorithm, as well as Greedy-HEU



(a) $N = 10$ (b) $S = 30$

FIGURE 6: Normalized revenue of the provider for different algorithms. The parameter N is the size of the alternative response time levels and S is the size of the submitted jobs.

method described earlier. The *normalized* total revenue is calculated as the percentile between the obtained results and the "exact optimal global result," which can be obtained by exhaustive search computing. Each result value is an average of 10 test cases. The detailed settings of our experiments are as follows.

First, the numbers of alternative job response time levels are fixed at 10, and the job population size varies from 10 to 50 with an increment of 10. The corresponding experimental results are shown in Figure 6(a). It can be seen that the GA optimization algorithm can find near-optimal solutions compared with "exact optimal global result," while Greedy-HEU cannot produce near-optimal results at most times. The result value of our GA optimization algorithm ranges between 90% and 100% when the job population size varies from 10 to 50, while the numbers of job response time levels are 10. It indicates that the size of the job population does not affect the quality of the proposed genetic optimization approach in obtaining the near-optimal solutions for the ICPRM problem.

Second, the job population size is fixed at 30, and the numbers of job execution time levels vary from 6 to 14 with an increment of 2. The results of this condition are shown in Figure 6(b). It can be seen that our GA can also find near-optimal solutions. The result value of our GA optimization algorithm ranges between 90% and 100% when the job response time levels vary from 6 to 14, while the job population size is 30. It indicates that the scale of job execution time levels does not affect the quality of the proposed GA in obtaining the near-optimal solutions for the revenue maximization problem.

5.4. Experimental Results and Comparison Discussion. For the ICPRM problem, Figure 6 demonstrates the normalized total revenue of the provider using our proposed GA and Greedy-HEU algorithms. It can be seen that the proposed approach can find near-optimal solutions, while Greedy-HEU cannot produce near-optimal results at most times. Though exhaustive search methods can help find exact global optimal solutions to the ICPRM problem, the CPU computational time needed by those methods becomes intractable and unacceptable with the growth of the problem instance size. For example, it takes about 20 minutes to get the global optimal solution for the ICPRM problem with the job population size being 10 and the number of job response time levels being 10. But when we increase the number of response time levels to 12, the exhaustive search method needs about 2 hours to get the optimal solution. Thus, the exhaustive search methods are not practicable and applicable to the cloud revenue maximization problem. Figures 4 and 5 report that for different problem instance sizes our proposed GA approach converges quickly and can find near-optimal solutions within 60 seconds. Above all, we can conclude that our proposed GA is both efficient and effective in finding near-optimal solutions for the ICPRM revenue maximization problems.

6. Related Work

The problem of revenue maximization in the cloud computing filed has become a crucial issue and attracted significant attention in the last few years. Below we provide a review of most relevant prior work.

Jin et al. propose a dynamic game to achieve Nash equilibrium in a distributed manner and tackle cloud price competition problem with Bertrand game modeling to maximize cloud provider's profit [5]. Their approach can help cloud providers compete for profit maximization in an oligopoly market. The work in [6] studies a cloud computing market where a cloud provider rents a set of computing resources from Windows Azure operated by Microsoft, and it proposes a stochastic programming model with two-stage recourse. The work in [4] aims at maximizing the revenues from SLA of multiclass systems where each class of request is assigned to a number of dedicated homogeneous servers to provide performance guarantees. The number of servers is evaluated by modeling each physical server as a G/G/1 queue, although the solution is determined by a very simple greedy approach.

Tsakalozos et al. propose a profit maximization approach based on microeconomics to direct the allotment of cloud resources for consumption in highly scalable master-worker virtual infrastructure [2]. Their proposed approach can maximize per-user financial profit and achieves resource sharing proportional to each user's budget. Zhang and Ardagna

propose a resource allocation controller for autonomic computing data center environments which maximizes the provider profits associated with multiclass Service Level Agreements [7]. Their main focus is on controlling the request routing and the processing sharing scheduling policies under predefined classes of jobs and utility-wise step functions. These approaches do not capture the differentiation of willingness-to-pay between clients and service demands are considered all equally, which is not realistic for IAAS environments described in this paper.

VM placement and migration approaches focusing on power optimization and profit maximization have received significant attention too and a number of related approaches have been proposed to address this issue [8–11]. Reference [12] proposes and evaluates energy-aware allocation policies that aim to maximize the average revenue received by the provider per unit time. A bin-packing based approach to maximize the resource satisfaction, minimize the number of application starts and stops, and balance the load across machines on a given datacenter is presented in [13]. The main focus of this work is on prompt response to changes in application demands but the problem of revenue issue is not investigated.

7. Conclusion and Future Work

In this paper, we argue that flexible provisioning of batch type job execution services in IAAS environments raises challenges not addressed by prior work on provisioning raw machine resources. We formulate provider's revenue maximization as an optimization problem, show its NP hardness, and develop a metaheuristic solution based on genetic algorithm. We illustrate the significance of proposed approach on a simulated environment. The experimental simulations and numerical results demonstrate that our proposed approach finds out near-optimal solutions to the revenue maximization problem under different problem sizes. The experimental results also show that the proposed GA optimization approach is sound on both efficiency and effectiveness for solving ICPRM problem.

For the future work, there are more related research issues worth investigating. First, we would like to extend our results to study solicitation of honest utility functions reporting by mechanisms design. Second, we are also interested in exploring pricing issues with multiple cloud providers, such as competition and peering.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors would like to thank the anonymous referees for their insightful comments that have improved the quality of the paper. This work is supported in part by the National Natural Science Foundation of China (61402263), the Natural Science Foundation of Shandong Province (ZR2014FQ031), and the Science & Technology Development Projects of Shandong Province (2014GGH201007), China.

References

[1] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 50–55, 2008.

[2] K. Tsakalozos, H. Kllapi, E. Sitaridi, M. Roussopoulos, D. Paparas, and A. Delis, "Flexible use of cloud resources through profit maximization and price discrimination," in *Proceedings of the IEEE 27th International Conference on Data Engineering (ICDE '11)*, pp. 75–86, IEEE, April 2011.

[3] S. Martello and P. Toth, "Algorithms for knapsack problems," *Surveys in Combinatorial Optimization*, vol. 31, pp. 213–258, 1987.

[4] B. Urgaonkar and P. Shenoy, "Sharc: managing CPU and network bandwidth in shared clusters," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 1, pp. 2–17, 2004.

[5] X. Jin, Y.-K. Kwok, and Y. Yan, "Competitive cloud resource pricing under a smart grid environment," in *Proceedings of the 5th IEEE International Conference on Cloud Computing Technology and Science (CloudCom '13)*, pp. 655–662, IEEE, December 2013.

[6] S. Chaisiri, B. Lee, and D. Niyato, "Competitive cloud resource pricing under a smart grid environment," in *Proceedings of the 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, pp. 1–4, 2012.

[7] L. Zhang and D. Ardagna, "SLA based profit optimization in autonomic computing systems," in *Proceedings of the 2nd International Conference on Service Oriented Computing (ICSOC '04)*, pp. 173–182, ACM, New York, NY, USA, November 2004.

[8] B. Li, J. Li, J. Huai, T. Wo, Q. Li, and L. Zhong, "EnaCloud: an energy-saving application live placement approach for cloud computing environments," in *Proceedings of the IEEE International Conference on Cloud Computing*, pp. 17–24, IEEE, September 2009.

[9] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," in *Proceedings of the 10th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid '10)*, pp. 826–831, May 2010.

[10] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755–768, 2012.

[11] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimal virtual machine placement across multiple cloud providers," in *Proceedings of the IEEE Asia-Pacific Services Computing Conference (APSCC '09)*, pp. 103–110, IEEE, Singapore, December 2009.

[12] M. Mazzucco, D. Dyachuk, and R. Deters, "Maximizing cloud providers' revenues via energy aware allocation policies," in *Proceedings of the 3rd IEEE International Conference on Cloud Computing*, pp. 131–138, IEEE, July 2010.

[13] C. Tang, M. Steinder, M. Spreitzer, and G. Pacifici, "A scalable application placement controller for enterprise data centers," in *Proceedings of the 16th International Conference on World Wide Web*, pp. 331–340, ACM, May 2007.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 658216, 10 pages
http://dx.doi.org/10.1155/2015/658216

Research Article

Wireless-Uplinks-Based Energy-Efficient Scheduling in Mobile Cloud Computing

Xing Liu,[1] Chaowei Yuan,[1] Zhen Yang,[2] and Enda Peng[1]

[1] *School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, No. 10 Xitucheng Road, Haidian, Beijing 100876, China*
[2] *School of Computer Science, Beijing University of Posts and Telecommunications, No. 10 Xitucheng Road, Haidian, Beijing 100876, China*

Correspondence should be addressed to Xing Liu; liuxing_bupt@qq.com

Received 16 November 2014; Accepted 2 February 2015

Academic Editor: Bao Rong Chang

Mobile cloud computing (MCC) combines cloud computing and mobile internet to improve the computational capabilities of resource-constrained mobile devices (MDs). In MCC, mobile users could not only improve the computational capability of MDs but also save operation consumption by offloading the mobile applications to the cloud. However, MCC faces the problem of energy efficiency because of time-varying channels when the offloading is being executed. In this paper, we address the issue of energy-efficient scheduling for wireless uplink in MCC. By introducing Lyapunov optimization, we first propose a scheduling algorithm that can dynamically choose channel to transmit data based on queue backlog and channel statistics. Then, we show that the proposed scheduling algorithm can make a tradeoff between queue backlog and energy consumption in a channel-aware MCC system. Simulation results show that the proposed scheduling algorithm can reduce the time average energy consumption for offloading compared to the existing algorithm.

1. Introduction

Mobile devices (MDs) are increasingly becoming an essential part of human life [1]. As the most effective and convenient tools, MDs are not bounded by time and place. However, the limited computational power, storage space, and battery lifetime of existing MDs significantly limit their ability to execute resource-intensive applications [2]. Fortunately, mobile cloud computing (MCC), which combines cloud computing with mobile internet, can provide service via Infrastructure as a Service (IaaS) platform [3]. Using this IaaS platform, on the one hand, the performance of MDs can be improved by offloading mobile applications to the cloud servers. On the other hand, as users offload the mobile applications to the cloud, the data, which are transmitted on both wired and wireless networks, are increased rapidly, the communication overhead of MD will significantly consume battery energy [4]. Therefore, reducing the energy consumption of transmission is one of the most significant issues in MCC.

The issue about energy efficiency of data transmission in wireless networks was investigated in [5–7]. In [5], Zafer and Modiano used a novel continuous-time optimal-control formulation and Lagrangian duality, to propose an optimal transmission scheduling that can dynamically adapt to the rate over time considering the channel variations to minimize the transmission energy cost. In [6], Fu and van der Schaar investigated the structure-aware online learning for the energy-efficient and delay-sensitive transmission. In [7], Neely exploited time-varying channel conditions to design an energy-efficient control (EEC) algorithm based on the Lyapunov optimization. However, all the above papers only consider a single user transmitting data over a time-varying channel. To this end, much research work [8, 9] has focused on the issue of energy-efficiency for multiuser multichannel

scenario. In [8], Li and Neely consider a wireless base station serving N users through N time-varying channels and propose a dynamic channel acquisition algorithm (DCAA) that all mobile users dynamically share all wireless channels. In [9], Xiang et al. used discrete-time stochastic dynamic program, to propose an approximate dynamic programming (ADP) that can dynamically select channel and execute data transmission scheduling with time-varying channel conditions. The basic idea of the proposed schemes is optimized energy-efficiency of downlink data transmission between MDs and base stations.

Although all of the above work focused on the wireless downlinks, the energy efficiency of the wireless uplink is lacking. In fact, the energy consumption of uploading data is more than that of downloading data [10]; hence reducing the energy consumption of uploading data is the key to save the energy consumption on MDs in the MCC system. In [11], Ra et al. implemented Lyapunov optimization framework on MDs in multiple wireless uplinks environments and designed a stable and adaptive link selection algorithm (SALSA) with time-varying V values for meeting different applications' delay tolerances. This SALSA algorithm can automatically select channels and requires channel state information to decide whether and when to defer a transmission. However, the SALSA algorithm supports each user with a dedicated-channel when it has already selected a channel while an application is executing. In this way, user might miss better transmission opportunity on other channels; it is not so good for reducing MDs' energy consumption. For example, there are two users (1 and 2) uploading data over two time-varying channels (A and B). Let SALSA algorithm allocate channel A and channel B to user 1 and user 2 at the beginning of uploading, respectively. We assume user 1 has low service rate on channel A and high service rate on channel B, while user 2 has low service rate on channel B and high service rate on channel A after a period of time. According to the SALSA algorithm, on the one hand, users may defer the data transfer until their channels' service rate is high enough, rather than reallocate channel. On the other hand, as the queue backlog increases, users may transmit the data to cloud with the low service rate. Indeed, the energy-delay tradeoff can be provided. However, the high energy consumption is incurred, because the SALSA algorithm cannot reallocate channel and users may upload data with the low service rate when the queue backlog is high enough.

In this paper, based on the MCC system with IaaS platform, we address the issue of energy-efficient scheduling considering the resource-intensive applications, which need to upload large data to the cloud such as mHealth, mobile commerce, and mobile office. In our scheduling algorithm, mobile user chooses appropriate channels to transmit applications' data packet with the queue backlog and channel statistics every T time unit. In the above example, our scheduling algorithm can transmit user 1's data over channel B when the service rate of channel B is higher than that of channel A. Compared with the SALSA algorithm, our scheduling algorithm has a lower queue backlog because it can transmit user's data over any channel that has a high service rate support user. Thus, for the same parameter V, which control the tradeoff between energy consumption and queue backlog, our scheduling algorithm has the lower energy consumption than SALSA algorithm. Our main contribution includes the following.

(i) Adopting the framework of Lyapunov optimization, we propose a two-time dynamic offloading (T2DO) algorithm to decrease the energy consumption on MDs by considering queue backlog and channel states.

(ii) We demonstrate that the proposed algorithm can approach the optimal energy consumption within $O(1/V)$ deviation, and a tradeoff in average queue backlog is $O(V)$.

(iii) We compare the performance of our T2DO algorithm with SALSA and random and minimum-delay algorithms using simulation. The results show that, by appropriately choosing V, the T2DO algorithm outperforms the other three algorithms, with smaller time average energy consumption while achieving queue stability.

The remainder of this paper is organized as follows. In Section 2, the review of related work is presented. In Section 3, we present the system model and problem statement in the MCC. In Section 4, according to Lyapunov optimization, we propose a T2DO algorithm which makes a tradeoff between queue backlog and energy consumption. The performance analysis of T2DO algorithm is given in Section 5. In Section 6, we compare the performance of the T2DO algorithm with SALSA and random and minimum-delay algorithms using simulation. Section 7 concludes this paper and provides future directions.

2. Related Work

MCC is an emerging technology to extend the capabilities of MDs by offloading. According to [12], not all applications offloading can extend battery life of MDs. Thus, a vast amount of previous work [10, 13–16] has been investigating the scheme of applications offloading. In [10], Altamimi et al. proposed an energy model for the task offloading to the cloud. It minimized power consumption on the MDs under the constraints of the responsiveness and accuracy of demands of interactive perception tasks. In [13], Chen formulated the decision problem of computation offloading among mobile users as a decentralized computation offloading game and proposed a computation offloading mechanism based on game-theoretic approach to save energy on the MD. In [14], Yang et al. discussed an assisting execution offloading approach by reducing the size of transferred state and proposed an execution offloading scheme to improve the performance of the MCC significantly in terms of execution time and energy consumption. In [15], Zhang et al. investigated collaborative task execution between MD and cloud clone for mobile applications under the stochastic wireless channel. In [16], Zhang et al. proposed an energy-optimal execution strategy for the MCC under the stochastic wireless channel. For the mobile execution, it minimized the computation

energy by dynamically configuring the clock frequency of the chip. For the cloud execution, it minimized the transmission energy by optimally scheduling data transmission across the stochastic wireless channel.

With offloading, the data transfer between MD and cloud is increasing rapidly; thus the higher data transmission energy consumption is incurred, especially with a bad wireless channel. Some literatures have studied the energy issues of data transmission for offloading. In [17], Kumar and Lu considered a fixed computation scheduling with a fixed data rate model in the MD for the wireless channel. In [18], Huang et al. presented a dynamic offloading algorithm to transfer data to save energy on the MD while meeting the application execution time. In [19], Tilevich and Kwon presented a determined functionality to offload at runtime, which can maximize efficiency by automating program transformation. In [9], Xiang et al. presented a flexible link selection and data transmission scheduling and proposed a scalable approximate dynamic programming (ADP) algorithm based on the discrete-time stochastic dynamic program to reduce the average energy consumed for delivering a packet.

In a real-world application, uplink power consumption dominates the wireless power budget due to RF power requirements for reliable transmission over long distances; thus energy-efficiency of uploading data is a key issue in MCC due to energy-poverty of MDs. Some researchers have worked on the energy-optimal scheduling in wireless uplinks. In [20], Katranaras et al. considered clustered cooperation and investigated effective techniques for managing uplinks of intercluster interference to improve users' performance in terms of both spectral and energy efficiency. In [21], Miao et al. developed an energy-efficient scheme with significantly lower complexity when compared to iterative approaches in an uplink OFDMA system. This scheme allocates the system bandwidth among all users to optimize energy efficiency across the whole network. In [22], Liu et al. proposed a dynamic carrier aggregation (DCA) scheme to improve the energy efficiency of uplink communications. In [23], Deb and Monogioudis proposed LeAP, a measurement data-driven machine learning paradigm for power control to manage uplink interference in LTE. The authors in [11] derived the energy-efficient scheduling automatically select channels and requires channel state information to decide whether and when to defer data uploading. However, it cannot reselect channels while an application is executing, even though the service rate of other channels is higher than that of current channel in some time.

This paper investigates energy-efficient scheduling for wireless uplinks in the MCC. Compared to previous works, this paper has several differences. First, we employ the multiqueueing model and consider MD can reselect appropriate channel to transmit data while an application is executing. Second, we provide the theoretical framework of data uploading, in which MD first calculates the amount of data transfer of each offloading application to balance queue length among all queues; then MD chooses appropriate channels to transmit its data according to the queue backlog and channel statistics every T time unit. Finally, we prove that the proposed scheduling achieves the exact $[O(1/V), O(V)]$ energy-queue tradeoff under wireless uplinks in the MCC system.

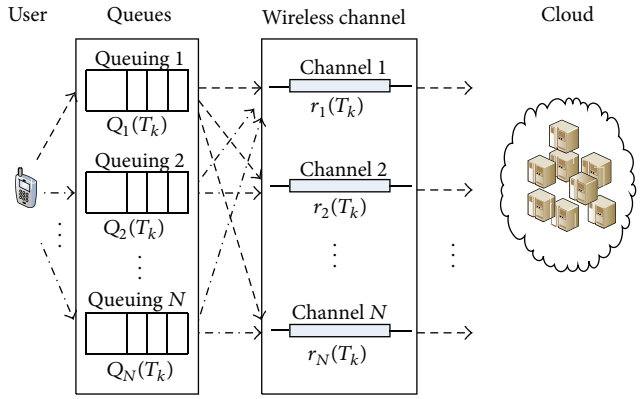


Figure 1: System model.

3. System Model and Problem Statement

In this section, we first present the model for resource-intensive application offloading in the MCC system. Then, we describe the formulation for the problem of energy-efficient for wireless uplinks. Finally, we employ the mHealth as an instance to show the process of the proposed algorithm.

3.1. System Model. We consider a MCC system with IaaS platform, which owns a server cluster to support mobile users. Without loss of generality, we use one server to represent the server cluster. When users are going to offload the mobile applications, the server will allocate appropriate amounts of virtual machines to execute these applications. For more details about this MCC system one can refer to [24–26]. Let T be the time unit; we use $T_k = \lceil t/T \rceil$ to denote the kth time unit, where t is any time. The MCC system operates in unit time. We consider a user who has N heterogeneous applications offloading to cloud through N time-varying channels. The data generated in each offloading application is processed in a corresponding queue, denoted by $\mathbf{Q}(T_k) = (Q_1(T_k), Q_2(T_k), \ldots, Q_N(T_k)), T_k = 0, 1, \ldots$, which is assumed to operate in a discrete time unit manner. For any application $i, Q_i(T_k)$ represents its queue backlog of data to be transmitted from the MD to the cloud at the beginning of the time unit T_k. Let $\mathbf{A}(T_k) = (A_1(T_k), A_2(T_k), \ldots, A_N(T_k))$ be the vector denoting the amount of newly generated data of each offloading application in every time unit T_k, which is referred to as the vector data arrival rates that arrive in their corresponding queues $\mathbf{Q}(T_k)$. Suppose $A_i(T_k)$ is Poisson distributed with $\mathbb{E}[A_i(T_k)] = \lambda_i$; then we have $\mathbb{E}[\mathbf{A}(T_k)] = \boldsymbol{\lambda} \triangleq (\lambda_1, \lambda_2, \ldots, \lambda_N)$. The system model is shown in Figure 1.

Let $\mathbf{r}(T_k) = (r_1(T_k), r_2(T_k), \ldots, r_N(T_k))$ be the vector of current channel states for different channels during time unit T_k. After acquiring channel states, MD chooses appropriate queues to transmit their data packet in each time slot τ, where $\tau = T/k_1, k_1 \in \mathbb{Z}^+$. Let $\mu_{ij}(T_k) \in \{0, 1, \ldots, \mu_{\max}\}$ be the service

rate of application i supported by channel j, where $\mu_{\max} \in \mathbb{Z}^+$. For the service rate of the queue i, we have

$$\mu_i(T_k) = \sum_{j=1}^{N} \mu_{ij}(T_k)\, w_{ij}(T_k), \tag{1}$$

where $w_{ij}(T_k)$ denotes the number of slots that the queue i is supported by channel j in time unit T_k.

According to [27], the power state of MD is divided into active state and idle state. Let $\alpha_i(T_k)$ be the power of the MD at the active state, which is supported by the channel j with current channel state $r_j(T_k)$ in time unit T_k. Let $\beta_i(T_k)$ be the idle power of the user in time unit T_k. Then the energy consumption of user in time unit T_k can be denoted as

$$E(T_k) = \sum_{i=1}^{N}\sum_{j=1}^{N}\left[\alpha_i(T_k)\, w_{ij}(T_k)\,\tau + \beta_i(T_k)\left(T - w_{ij}(T_k)\,\tau\right)\right], \tag{2}$$

where $\sum_{j=1}^{N} w_{ij}(T_k)\tau \le T$.

3.2. Problem Statement. In the following, we assume that MD can estimate the unfinished delivery in its queues accurately. $\mathbf{Q}(T_k) = (Q_1(T_k), Q_2(T_k), \ldots, Q_N(T_k))$ is the unfinished application data of the offloading in the MD at time unit T_k. We use the following queueing dynamics

$$Q_i(T_k + 1) = \max\left[Q_i(T_k) - \mu_i(T_k), 0\right] + A_i(T_k). \tag{3}$$

Throughout the paper, we require all the queues to be stable, which is defined as

$$\overline{Q} \triangleq \limsup_{T_k \to \infty} \frac{1}{T_k} \sum_{s=1}^{T_k} \sum_{i=1}^{N} \mathbb{E}\left\{Q_i(s)\right\} < \infty, \tag{4}$$

where $\overline{Q}$ is the time average queue length and the expectation is taken over the randomness of $\mathbf{Q}(T_k)$.

In practice, dead spots or coverage holes usually cause disconnection (i.e., WLAN and 3G are not available). In this situation, a huge queue backlog even queue instability might be caused, because the mobile application cannot be offloaded to cloud any more. In fact, this problem can be tackled by making the cloud services temporarily stopped. We will treat this case as a joint optimization of local execution and transmitting cost in the future. Moreover, MCC faces the channel's competition problem, when several users offload the mobile applications to cloud at the same time. This problem has been treated in [28, 29] and is beyond the scope of this paper. In this paper, we only consider that a mobile user offloads mobile application from MD to cloud.

The focus of our work is energy optimal scheduling for data uploading with the time-varying wireless channel. We call every *feasible* policy that ensures (4) a *stable* policy and use E_{av}^{*} to denote the infimum average energy consumption over all stable policies. Then, we define the time average energy consumption of a feasible policy Π as

$$E_{\text{av}}^{\Pi} \triangleq \limsup_{T_k \to \infty} \frac{1}{T_k} \sum_{s=1}^{T_k} \mathbb{E}\left\{E^{\Pi}(s)\right\}, \tag{5}$$

where $E^{\Pi}(s)$ denotes the energy consumption on the MD by policy Π at time unit T_k.

The objective is to find a stable policy by reducing the number of transmission slots at every time unit T_k, so as to minimize the time average energy consumption of MD. We refer to this as the energy consumption minimization (ECM) problem in the remainder of the paper.

4. A Two-Time Dynamic Offloading Algorithm

In this section, we first analyze the ECM problem using Lyapunov optimization. Then, we describe T2DO algorithm and discuss corresponding insight and implementation-related issues.

4.1. The ECM Problem Analysis Using Lyapunov Optimization. We first define the Lyapunov function, $L(T_k)$, to measure the aggregate queue backlog in the system

$$L(T_k) \triangleq \sum_{i=1}^{N} \frac{1}{2}\left(\left[Q_i(T_k)\right]^2\right). \tag{6}$$

Next, we define the T-unit Lyapunov drift, $\Delta_T(T_k)$ as the expected change in the Lyapunov function over T_k units

$$\Delta_T(T_k) \triangleq \mathbb{E}\left\{L(T_k + 1) - L(T_k) \mid \mathbf{Q}(T_k)\right\}. \tag{7}$$

Following the Lyapunov optimization approach, we add the expected energy consumption with $V > 0$ over T_k units (i.e., a penalty function), $V\mathbb{E}\{E(T_k)\}$, to (7), which leads to the *drift-plus-penalty* term. This is a key step to obtain an upper bound on this term. Before further discussing the *drift-plus-penalty* term, we first present a lemma in [30], which is related to the derivation of the upper bound of *drift-plus-penalty* term.

Lemma 1. *If $A, B, C, D \in \mathbb{R}^+$, and $A \le \max[B - C, 0] + D$, then $A^2 \le B^2 + C^2 + D^2 - 2B(C - D)$.*

Based on (7) and Lemma 1, we derive a theorem, which characterizes the upper bound for our case.

Theorem 2. *For any given $V > 0$, under any possible actions $\mu_{ij}(T_k) \in \{0, 1, \ldots, \mu_{max}\}$ and $A_i(T_k) \in \{0, 1, \ldots, A_{max}\}$, one has*

$$\begin{aligned}\Delta_T(T_k) + V\mathbb{E}\left\{E(T_k) \mid \mathbf{Q}(T_k)\right\} &\le B + V\mathbb{E}\left\{E(T_k) \mid \mathbf{Q}(T_k)\right\} \\ &\quad - \sum_{i=1}^{N} \mathbb{E}\left\{Q_i(T_k)\left[\mu_i(T_k) - A_i(T_k)\right] \mid \mathbf{Q}(T_k)\right\},\end{aligned} \tag{8}$$

where $B = (N^3\mu_{max}^2 + NA_{max}^2)/2$.

Input: $\mathbf{Q}(T_k)$
Output: $\mu_i(T_k)$
(1) At the beginning of time unit T_k, monitor the queue backlog.
(2) Monitor the channel states $\mathbf{r}(T_k)$ between cloud and MD.
(3) The mobile agent chooses queue $Q_i(T_k)$ to transmit data to minimize
$$|Q_i(T_k) - Q_j(T_k)| \quad i \neq j, 0 \leq i, j \leq N \qquad (*)$$
(4) **if** $VE(T_k) - \sum_{i=1}^{N} Q_i(T_k)\mu_i(T_k) < 0$ **then**
(5) The mobile agent chooses the channels to transmit data to minimize
$$VE(T_k) - \sum_{i=1}^{N} Q_i(T_k)\mu_i(T_k) \qquad (**)$$
(6) **else**
(7) Stay idle for energy conservation.
(8) **end if**
(9) Update the queues using (1).

Algorithm 1: T2DO algorithm.

Proof. Squaring both sides of the queueing dynamic (1) and using Lemma 1, we have

$$\begin{aligned}\left[Q_i\left(T_k+1\right)\right]^2 \leq \left[Q_i\left(T_k\right)\right]^2 + \left[\mu_i\left(T_k\right)\right]^2 + \left[A_i\left(T_k\right)\right]^2 \\ - 2Q_i\left(T_k\right)\left[\mu_i\left(T_k\right) - A_i\left(T_k\right)\right].\end{aligned} \qquad (9)$$

Summing (9) from $i = 1$ to N, using the fact that $\mu_i(T_k) \leq N\mu_{\max}$ and $A_i(T_k) \leq A_{\max}$, we obtain

$$\begin{aligned}&\sum_{i=1}^{N}\left(\left[Q_i\left(T_k+1\right)\right]^2 - \left[Q_i\left(T_k\right)\right]^2\right) \\ &\leq N^3\mu_{\max}^2 + NA_{\max}^2 - 2\sum_{i=1}^{N} Q_i\left(T_k\right)\left[\mu_i\left(T_k\right) - A_i\left(T_k\right)\right].\end{aligned} \qquad (10)$$

Substituting (10) into (7), we get

$$\begin{aligned}\Delta_T\left(T_k\right) &\triangleq \mathbb{E}\left\{L\left(T_k+1\right) - L\left(T_k\right) \mid \mathbf{Q}\left(T_k\right)\right\} \\ &\leq \sum_{i=1}^{N}\mathbb{E}\left\{Q_i\left(T_k\right)\left[A_i\left(T_k\right) - \mu_i\left(T_k\right)\right] \mid \mathbf{Q}\left(T_k\right)\right\} \\ &\quad + \frac{1}{2}\left(N^3\mu_{\max}^2 + NA_{\max}^2\right).\end{aligned} \qquad (11)$$

Therefor by defining $B = (N^3\mu_{\max}^2 + NA_{\max}^2)/2$ and adding $V\mathbb{E}\{E(T_k) \mid \mathbf{Q}(T_k)\}$ to both sides of (11), we obtain (8). □

4.2. The T2DO Algorithm Design (See Algorithm 1). According to [18], the design principle of Lyapunov framework is minimizing the upper bound of the *drift-plus-penalty* term; that is, in every time unit T_k, we try to choose a scheduling algorithm to minimize the RHS of (8). In fact, the scheduling algorithm only affects the energy consumption $E(T_k)$ and the queue i's service rate $\mu_i(T_k)$ in the time unit T_k; hence we can minimize the RHS of (8) by minimizing the following simplified term

$$V\mathbb{E}\left\{E\left(T_k\right) \mid \mathbf{Q}\left(T_k\right)\right\} - \sum_{i=1}^{N}\mathbb{E}\left\{Q_i\left(T_k\right)\mu_i\left(T_k\right) \mid \mathbf{Q}\left(T_k\right)\right\}. \qquad (12)$$

We now describe the system implementation of T2DO algorithm as follows.

The T2DO algorithm works at two different time scales. The MD assigns the service rate of application offloading at the beginning of every time unit T. Then MD chooses the channel to transmit data at every time τ slot. Two different time scales are important from an implementation perspective, because the MD's decision interval is usually much longer than channels' data transmission slot τ.

The proposed T2DO algorithm has two important properties. First, MD assigning decision remains unchanged for a time unit. We can properly increase T to reduce the computational overhead at the MD. Second, channel can defer the transmission of data if its channel state is too bad. Channel i may choose not to transmit data in a particular time slot τ or time unit T, even if $Q_i(T_k) > 0$, due to the low data transmission rates at the user.

4.3. A Practical Instance for T2DO Algorithm. In this section, we take the mHealth as an instance to show the process of the T2DO algorithm. The mHealth combines wearable body sensors [31] and MD (i.e., mobile-based medical monitoring device) to provide more effective and affordable healthcare services. Medical organizations focusing on hypertension, diabetes, geriatrics, or chronic diseases can take care of patients at their home instead of in the hospital. However, one of the disadvantages of eHealth is that different medical technologies are often supported by different vendors, which leads to poor interoperability. In addition, dedicated MDs are required. Constrained by computational power, storage space, and battery life of existing MDs, it is hard for them to execute resource-intensive applications. In this case, as an

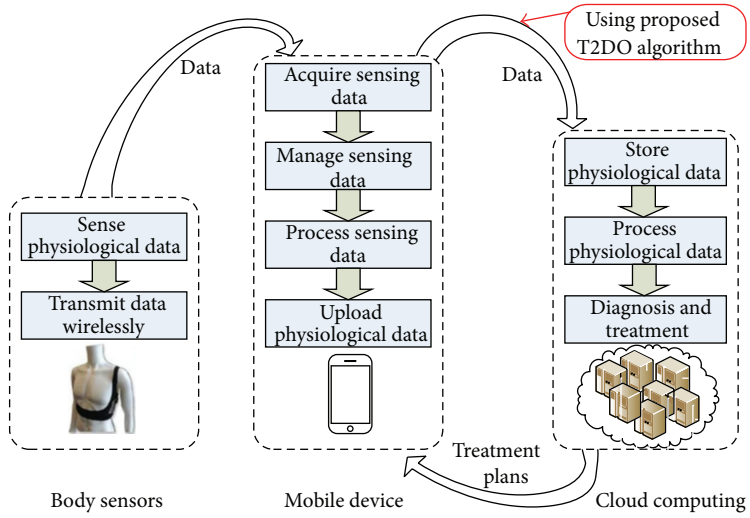


Figure 2: MCC-based mHealth.

IaaS platform, cloud computing can contribute to offloading the eHealth services from MDs to the cloud. This paradigm not only improves the performance and the compatibility of MDs, but also raises the possibility of providing more accurate offsite personalized medical diagnosis and treatment [32].

In a MCC-based mHealth, as shown in Figure 2, the MD first connects to physiological body sensors to collect data, such as blood pressure, temperature, heart rate, and electrocardiograph. Then, the mobile device manages all sensing data acquired from wearable body sensors and uploads the physiological data to the cloud. Finally, the cloud stores and processes these physiological data. Once the diagnostic decision on the cloud is finished, the treatment plans will be sent back to the MD. In this case, our T2DO algorithm can reduce the energy consumed in data uploading between the MD and the cloud. This part of energy is the major energy consumption on MD for the mHealth.

Next, we show that how the T2DO algorithm reduces the energy consumption of uploading for the mHealth. Here, we also employ the mobile agent as an entity to manage and upload physiological data; thus the mobile agent is an executor of our T2DO algorithm. When the MD obtains physiological data from wearable body sensors, the mobile agent firstly estimates the queue backlog of these physiological data and monitors the channel state between the MD and the cloud at the beginning of time unit T_k. Secondly, the mobile agent chooses queue $Q_i(T_k)$ to transmit data to balance queue length among all queues. Thirdly, the mobile agent decides whether the data of queues be transmitted or not in current time unit and decides the amount of data of those queues which can be transmitted based on equality $(**)$. Finally, according to the amount of transmitting and arriving data, the mobile agent updates the queues based on equality (3). According to above process of the T2DO algorithm, we can save the energy consumption on MD by postponing data transmission when the channel states are terrible.

In this paper, we only take the mHealth as an example to show how our T2DO algorithm reduces the energy consumption of uploading. In fact, the T2DO algorithm can also be applied to other mobile applications such as mobile commerce, mobile learning, and mobile office.

5. Performance Analysis

According to [7], we characterize the optimal time average energy consumption E_{av}^* with below lemma, which can be achieved by any algorithm that stabilizes the queue.

Lemma 3. *For any data arrival rate vector $\boldsymbol{\lambda} \in \Lambda$ and time unit T_k, there exists a stationary randomized control policy Π_{opt} that chooses appropriate queue to transmit its data packet in each slot τ; then one has the following equalities*

$$\begin{aligned} \mathbb{E}\left\{E^{\Pi_{opt}}\left(T_k\right)\right\} &= E_{av}^*\left(\boldsymbol{\lambda}\right) \\ \mathbb{E}\left\{\mathbf{A}\left(T_k\right)\right\} &= \mathbb{E}\left\{\sum_{i=1}^{N} \mu_i^{\Pi_{opt}}\left(T_k\right)\right\}, \end{aligned} \tag{13}$$

where Λ denotes the capacity region of the system.

Lemma 3 shows that by using a stationary randomized algorithm, it is possible to achieve the minimum time average energy consumption E_{av}^* for a given data arrival rate vector $\boldsymbol{\lambda}$.

Based on Theorem 2 and Lemma 3, we derive a theorem, which presents bounds on the time average energy consumption and queue backlogs achieved by T2DO algorithm.

Theorem 4. *Suppose there exists an $\epsilon > 0$ such that $\boldsymbol{\lambda}+\epsilon\mathbf{1} \in \Lambda$; then under T2DO algorithm, the performance bounds of the*

time average energy consumption and queue backlog can be denoted as

$$\overline{Q^{T2DO}} \triangleq \limsup_{K\to\infty} \frac{1}{K} \sum_{T_k=1}^{K} \sum_{i=1}^{N} \left(\mathbb{E}\left\{Q_i\left(T_k\right)\right\}\right) \leq \frac{B + VE_{max}}{\epsilon} \quad (14)$$

$$\overline{E^{T2DO}} \triangleq \limsup_{K\to\infty} \frac{1}{K} \sum_{T_k=1}^{K} \mathbb{E}\left\{E^{T2DO}\left(T_k\right)\right\} \leq E_{av}^{*}\left(\boldsymbol{\lambda}\right) + \frac{B}{V}, \quad (15)$$

where $\mathbf{1}$ *denotes the vector of all 1's and* E_{av}^{*} *and* E_{max} *are the optimal energy consumption and the maximum energy consumption for stationary randomized control, respectively.*

Proof. Since $\boldsymbol{\lambda} + \epsilon\mathbf{1} \in \Lambda$, it can be shown using Lemma 3 that there exists a stationary and randomized policy Π'_{opt} that achieves the following

$$\begin{aligned} \mathbb{E}\left\{E^{\Pi'_{\text{opt}}}\left(T_k\right)\right\} &= E_{\text{av}}^{*}\left(\boldsymbol{\lambda} + \epsilon\mathbf{1}\right) \\ \mathbb{E}\left\{\mathbf{A}\left(T_k\right)\right\} &= \mathbb{E}\left\{\sum_{i=1}^{N} \mu_i^{\Pi'_{\text{opt}}}\left(T_k\right)\right\} - \epsilon, \quad \forall i, \end{aligned} \quad (16)$$

where $E_{\text{av}}^{*}(\boldsymbol{\lambda} + \epsilon\mathbf{1})$ is the optimal energy consumption corresponding to the rate vector $\boldsymbol{\lambda} + \epsilon\mathbf{1} \in \Lambda$. Substituting (16) into (8), we get

$$\begin{aligned} &\Delta_T\left(T_k\right) + V\mathbb{E}\left\{E^{\text{T2DO}}\left(T_k\right) \mid \mathbf{Q}\left(T_k\right)\right\} \\ &\quad \leq B + VE_{\text{av}}^{*}\left(\boldsymbol{\lambda} + \epsilon\mathbf{1}\right) - \epsilon\left(\mathbb{E}\left\{\sum_{i=1}^{N} Q_i\left(T_k\right) \mid \mathbf{Q}\left(T_k\right)\right\}\right). \end{aligned} \quad (17)$$

Taking the expectation of (17) over $\mathbf{Q}(T_k)$, we obtain

$$\begin{aligned} &\mathbb{E}\left\{L\left(T_k + 1\right) - L\left(T_k\right)\right\} + \text{V}\mathbb{E}\left\{E^{\text{T2DO}}\left(T_k\right)\right\} \\ &\quad \leq B + VE_{\text{av}}^{*}\left(\boldsymbol{\lambda} + \epsilon\mathbf{1}\right) - \epsilon\left(\mathbb{E}\left\{\sum_{i=1}^{N} Q_i\left(T_k\right)\right\}\right). \end{aligned} \quad (18)$$

Summing (18) from $T_k = 0$ to K, we have

$$\begin{aligned} &\mathbb{E}\left\{L\left(T_k + 1\right)\right\} + V\mathbb{E}\left\{E^{\text{T2DO}}\left(T_k\right)\right\} \\ &\quad \leq KB + KVE_{\text{av}}^{*}\left(\boldsymbol{\lambda} + \epsilon\mathbf{1}\right) - \epsilon\sum_{T_k=1}^{K}\left(\mathbb{E}\left\{\sum_{i=1}^{N} Q_i\left(T_k\right)\right\}\right). \end{aligned} \quad (19)$$

Then, using the fact that $E_{\text{av}}^{*}(\boldsymbol{\lambda}+\epsilon\mathbf{1}) < E_{\max}$, $\mathbb{E}\{L(T_k+1)\} > 0$, and $V\mathbb{E}\{E^{\text{T2DO}}(T_k)\} > 0$ we have

$$\epsilon\sum_{T_k=1}^{K}\left(\mathbb{E}\left\{\sum_{i=1}^{N} Q_i\left(T_k\right)\right\}\right) \leq KB + KVE_{\text{av}}^{*}\left(\boldsymbol{\lambda} + \epsilon\mathbf{1}\right). \quad (20)$$

Dividing both sides of (20) by ϵK, we obtain

$$\frac{1}{K}\sum_{T_k=1}^{K}\left(\mathbb{E}\left\{\sum_{i=1}^{N} Q_i\left(T_k\right)\right\}\right) \leq \frac{B + VE_{\max}}{\epsilon}. \quad (21)$$

Taking a lim sup as $K \to \infty$, (14) is proven.

Moreover, based on (19), using the fact that $\mathbb{E}\{L(T_k+1)\} > 0$ and $\mathbb{E}\{\sum_{i=1}^{N} Q_i(T_k)\} > 0$, we can derive

$$V\mathbb{E}\left\{E^{\text{T2DO}}\left(T_k\right)\right\} \leq B + VE_{\text{av}}^{*}\left(\boldsymbol{\lambda} + \epsilon\mathbf{1}\right). \quad (22)$$

Summing (22) from $T_k = 1$ to K and dividing both sides by KV, we have

$$\frac{1}{K}\mathbb{E}\left\{\sum_{T_k=1}^{K} E^{T2S2}\left(T_k\right)\right\} \leq E_{\text{av}}^{*}\left(\boldsymbol{\lambda} + \epsilon\mathbf{1}\right) + \frac{B}{V}. \quad (23)$$

Taking a lim sup as $K \to \infty$, using Lebesgue's dominated convergence theorem, and then letting $\epsilon \to 0$, we obtain inequality (15). □

Equation (15) demonstrates that the energy consumption $\overline{E^{\text{T2DO}}}$ can arbitrarily close to the optimum $E_{\text{av}}^{*}(\boldsymbol{\lambda})$ with an arbitrarily large V (i.e., approach $E_{\text{av}}^{*}(\boldsymbol{\lambda})$ within $O(1/V)$ deviation). At the same time, according to (14), the proposed algorithm guarantees an $O(V)$ average queue backlog. Thus, by appropriately selecting the control parameter V, we can achieve a desired tradeoff between energy consumption and queue backlog.

From the perspective of algorithm's performance, it is very important to value the parameter V. The following theorem gives the range of values for V.

Theorem 5. *Assume that* $\mu_{ij}(T_k) \in \{0, 1, \ldots, \mu_{max}\}$ *is the service rate of application* i *supported by channel* j*, where* $\mu_{max} \in \mathbb{Z}^{+}$*,* E_T *is the energy consumption of offload application during the time unit* T_k*, and the reasonable range of values allowed for* V *is* $(0, M^2\mu_{max}^2/E_T)$*, where* $M = T/\tau$*.*

Proof. Suppose there exists V', MD transmits queue i's data in the time unit T_k, and the service rate of the queue i is $\mu_i(T_k)$. According to the T2DO algorithm, we have

$$V'E_T - \sum_{i=1}^{N} Q_i\left(T_k\right)\mu_i\left(T_k\right) < 0. \quad (24)$$

Let some channel j be in the best state, its corresponding service rate $\mu_j(T_k) = (T/\tau)\mu_{\max}$, and we are setting $M = T/\tau$. If $\sum_{i=1}^{N} Q_i(T_k) < M\mu_{\max}$, then MD may defer a transmission for saving energy on the MD. Otherwise, MD chooses transmitting data. In this case, based on (24), we have

$$V'E_T - \sum_{i=1}^{N} Q_i\left(T_k\right)M\mu_{\max} \leq V'E_T - M\mu_{\max}M\mu_{\max} < 0. \quad (25)$$

Based on (25), we can derive $V' < M^2\mu_{\max}^2/E_T$. Moreover, according to Lyapunov optimization approach, we have $V' > 0$. Thus, we may safely draw the conclusion that the reasonable range of values allowed for V is $(0, M^2\mu_{\max}^2/E_T)$. □

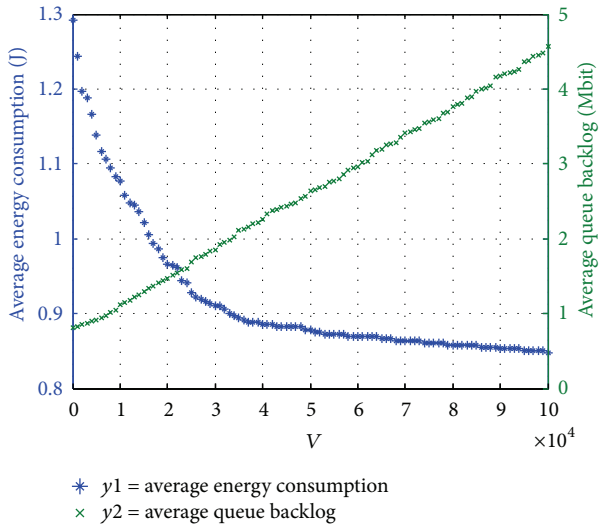


FIGURE 3: The impact of V on time average energy consumption and queue backlog.

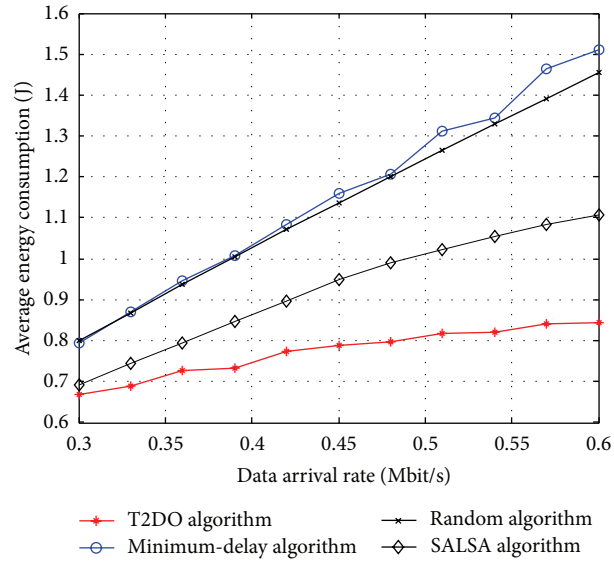


FIGURE 4: Average energy consumption comparison among different algorithms.

6. Simulation Results

In this section, we take the mHealth as a data uploading scenario to evaluate the performance of the proposed algorithm, as shown in Figure 3. Let each MD upload physiological data to the cloud with four time-varying channels. According to [27], the powers of MD at active state and idle state are set to 1.680 W and 0.594 W, respectively. The average service rate of each channel is set to 1000 Kbit/s. The time unit T is set to 1 s; the slot τ is set to 0.1 s. We assume the data arrival rate of each application equals 800 Kbit/s. The simulation time K is set to 1000 s. Each Q_i is assumed to have an empty queue at the first time unit.

We first characterize the tradeoff between the energy consumption and the queue backlog for the T2DO algorithm. We plot the tradeoff between energy consumption and queue backlog for data transfer in Figure 3. It is clear that energy consumption falls quickly at the beginning and then tends to descend slowly while the time average queue backlog grows linearly with V. Hence, the variable V controls the energy-delay tradeoff of the T2DO algorithm. These results in Figure 3 are consistent with Theorem 4. Particularly, there exists a sweet spot of V (e.g., $V = 40000$), beyond which increasing V leads to little energy conservation yet significantly increases average queue backlogs.

Next, let V be fixed to 40000; we vary the data arrival rate from 0.3 Mbit/s to 0.6 Mbit/s. Figure 4 shows a comparison of energy consumption among our T2DO algorithm, random algorithm, SALSA algorithm, and minimum-delay algorithm. Comparing with the random algorithm and the minimum-delay algorithm, the energy consumption of the proposed T2DO algorithm has decreased by 30%. The reason is that our T2DO algorithm can find an expected channel with good condition to transmit the data by postponing the communication for a while. Furthermore, we also note that proposed T2DO algorithm can provide better performance than SALSA algorithm. This is because user can obtain data from any channel with better channel conditions in the proposed algorithm, while in SALSA algorithm user is only supported by a dedicated-channel.

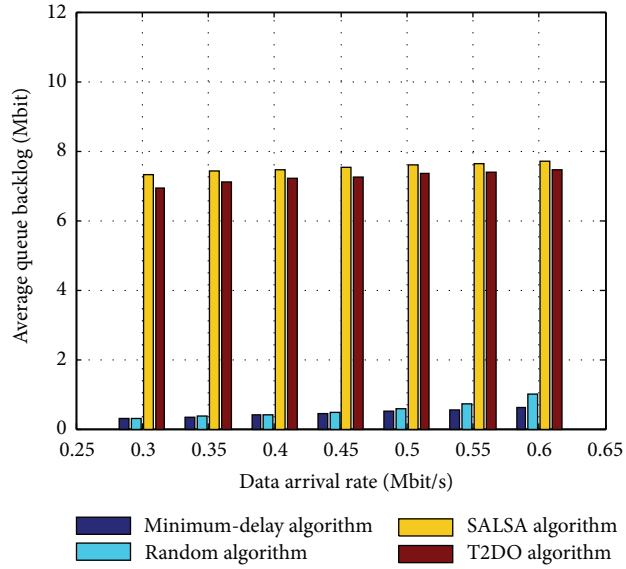


FIGURE 5: Average queue backlog among different algorithms.

Figure 5 shows a comparison of average queue backlog among four algorithms. We can see that the average queue backlog of proposed T2DO algorithm is longer than that of random algorithm and minimum-delay algorithm. It verifies that proposed T2DO algorithm can provide energy-delay tradeoff. Furthermore, we also note that proposed T2DO algorithm can provide better performance than SALSA algorithm in queue backlog. This is because the proposed T2DO algorithm can effectively increase service rate by dynamically

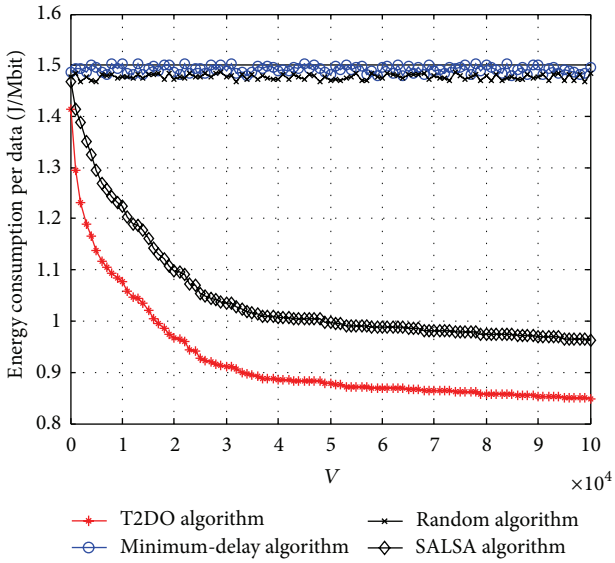


FIGURE 6: The $\overline{E_e}$ comparison among different algorithms.

choosing appropriate channels to transmit data for each time unit, and higher service rate results in lower queue backlog.

Finally, we compare the energy efficiency of T2DO algorithm with that of random algorithm, SALSA algorithm, and minimum-delay algorithm. For ease of comparison, we define a performance metric E_e, referred to as the average energy consumption per data packet

$$\overline{E_e} = \lim_{T\to\infty} \frac{\sum_{k=0}^{T} E(T_k)}{T\sum_{i=1}^{N}\lambda_i}. \tag{26}$$

Let the data arrival rate be fixed to 0.9 Mbit/s, and let the simulation time K be set to 1000 s. we vary V from 0 to 100000. Figure 6 compares the values of $\overline{E_e}$ under the four algorithms. It shows that the random algorithm and minimum-delay algorithm consume about 1.5 Joules to transmit per Mbit data. We can see that the $\overline{E_e}$ of T2DO algorithm and SALSA algorithm is lower than that of random algorithm and minimum-delay algorithm. Since T2DO algorithm and SALSA algorithm are designed according to Lyapunov optimization, they can transmit the data with good condition by postponing the communication for a while. It is interesting to note that the $\overline{E_e}$ of T2DO algorithm and SALSA algorithm tends to decrease as the parameter V increases, and T2DO algorithm can provide better performance than SALSA algorithm. The reason is that the parameter V control the tradeoff between energy consumption and queue backlog, so the energy consumption will decrease when V increases. Moreover, T2DO algorithm can effectively reduce the queue backlog by transmitting data over any channel with better channel conditions; thus for the same parameter V, our T2DO algorithm has the lower energy consumption per data than SALSA algorithm.

7. Conclusion

In this paper, we investigate the energy-efficient problem about the uploading data when applications are offloaded with a time-varying channel scenario in MCC. By the Lyapunov optimization, a T2DO algorithm is presented, where the tradeoff between energy and queue backlog for offloading is achieved. This T2DO algorithm first allocates data to the appropriate queuing and then chooses the appropriate channel to transmit their data packet with the queue backlog and channel statistics every T time unit. Moreover, we take the mHealth as an example and show the process of the proposed algorithm in real applications. The simulation results fully demonstrate the correctness and effectiveness of the proposed algorithm.

Unfortunately, this study also suffers from some practical issues. For instance, the delay constraint is another key factor of concern for mobile application, besides computational performance and energy efficiency. Moreover, different mobile applications have delay requirements. In the follow-up work, we will characterize the energy consumption caused by various applications uploading and explore alternative ways to reduce transmission energy, while meeting their delay requirements. we also plan to evaluate the performance of our T2DO algorithm by running real mobile applications, such as mHealth, mobile office, mobile commerce, and mobile learning.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Nature Science Foundation of China under Grant no. 61173017 and the National High-Tech R&D Program under Grant no. 2014AA01A701.

References

[1] F. Liu, P. Shu, H. Jin et al., "Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 14–21, 2013.

[2] C. Ge, Z. Sun, and N. Wang, "A survey of power-saving techniques on data centers and content delivery networks," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1334–1354, 2013.

[3] H. B. Liang, L. X. Cai, D. J. Huang, X. M. Shen, and D. Y. Peng, "An SMDP-based service model for interdomain resource allocation in mobile cloud networks," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 5, pp. 2222–2232, 2012.

[4] M. Shiraz, A. Gani, R. H. Khokhar, and R. Buyya, "A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1294–1313, 2013.

[5] M. Zafer and E. Modiano, "Minimum energy transmission over a wireless channel with deadline and power constraints," *IEEE*

Transactions on Automatic Control, vol. 54, no. 12, pp. 2841–2852, 2009.

[6] F. W. Fu and M. van der Schaar, "Structure-aware stochastic control for transmission scheduling," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 9, pp. 3931–3945, 2012.

[7] M. J. Neely, "Energy optimal control for time-varying wireless networks," *IEEE Transactions on Information Theory*, vol. 52, no. 7, pp. 2915–2934, 2006.

[8] C.-P. Li and M. J. Neely, "Energy-optimal scheduling with dynamic channel acquisition in wireless downlinks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 4, pp. 527–539, 2010.

[9] X. Xiang, C. Lin, and X. Chen, "Energy-efficient link selection and transmission scheduling in mobile cloud computing," *IEEE Wireless Communications Letters*, vol. 3, no. 2, pp. 153–156, 2014.

[10] M. Altamimi, A. Abdrabou, S. Naik, and A. A. Nayak, "Energy cost models of smartphones for task offloading to the cloud," *IEEE Transactions on Emerging Topics Computing*, vol. 6, no. 1, pp. 1–14, 2015.

[11] M.-R. Ra, J. Paek, A. B. Sharma, R. Govindan, M. H. Krieger, and M. J. Neely, "Energy-delay tradeoffs in smartphone applications," in *Proceedings of the 8th Annual International Conference on Mobile Systems, Applications and Services (MobiSys '10)*, pp. 255–269, San Francisco, Calif, USA, June 2010.

[12] U. Mandal, M. Habib, S. Zhang, B. Mukherjee, and M. Tornatore, "Greening the cloud using renewable-energy-aware service migration," *IEEE Network*, vol. 27, no. 6, pp. 36–43, 2013.

[13] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 5, pp. 1045–1057, 2014.

[14] S. Yang, D. Kwon, H. Yi, Y. Cho, Y. Kwon, and Y. Paek, "Techniques to minimize state transfer costs for dynamic execution offloading in mobile cloud computing," *IEEE Transactions on Mobile Computing*, vol. 13, no. 11, pp. 2648–2660, 2014.

[15] W. Zhang, Y. Wen, and D.-O. Wu, "Collaborative task execution in mobile cloud computing under a stochastic wireless channel," *IEEE Transactions on Wireless Communications*, vol. 14, no. 1, pp. 81–93, 2015.

[16] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, "Energy-optimal mobile cloud computing under stochastic wireless channel," *IEEE Transactions on Wireless Communications*, vol. 12, no. 9, pp. 4569–4581, 2013.

[17] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: can offloading computation save energy?" *IEEE Computer*, vol. 43, no. 4, pp. 51–56, 2010.

[18] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 1991–1995, 2012.

[19] E. Tilevich and Y. W. Kwon, "Cloud-based execution to improve mobile application energy efficiency," *Computer*, vol. 47, no. 1, pp. 75–77, 2014.

[20] E. Katranaras, M. A. Imran, M. Dianati, and R. Tafazolli, "Green inter-cluster interference management in uplink of multi-cell processing systems," *IEEE Transactions on Wireless Communications*, vol. 13, no. 12, pp. 6580–6592, 2014.

[21] G. Miao, N. Himayat, G. Y. Li, and S. Talwar, "Low-complexity energy-efficient scheduling for uplink OFDMA," *IEEE Transactions on Communications*, vol. 60, no. 1, pp. 112–120, 2012.

[22] F. Liu, K. Zheng, W. Xiang, and H. Zhao, "Design and performance analysis of An energy-efficient uplink carrier aggregation scheme," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 2, pp. 197–207, 2014.

[23] S. Deb and P. Monogioudis, "Learning-based uplink interference management in 4G LTE cellular systems," *IEEE/ACM Transactions on Networking*, vol. 11, no. 9, pp. 1063–1081, 2014.

[24] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: taxonomy and open challenges," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 369–392, 2014.

[25] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, "Cloud-based augmentation for mobile devices: motivation, taxonomies, and open challenges," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 337–368, 2014.

[26] P. Makris, D. N. Skoutas, and C. Skianis, "A survey on context-aware mobile and wireless networking: on networking and computing environments' integration," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 362–386, 2013.

[27] S. Berger, B. Almeroth, V. Suryaprakash, P. Zanier, I. Viering, and G. Fettweis, "Dynamic range-aware uplink transmit power control in LTE networks: establishing an operational range for LTE's open-loop transmit power control parameters (α, P_0)," *IEEE Wireless Communications Letters*, vol. 3, no. 5, pp. 521–524, 2014.

[28] R. Kaewpuang, D. Niyato, P. Wang, and E. Hossain, "A framework for cooperative resource management in mobile cloud computing," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 2685–2700, 2013.

[29] B. Cao, G. Feng, Y. Li, and C. Wang, "Cooperative media access control with optimal relay selection in error-prone wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 1, pp. 252–265, 2014.

[30] L. Huang, S. Moeller, M. J. Neely, and B. Krishnamachari, "LIFO-backpressure achieves near-optimal utility-delay tradeoff," *IEEE/ACM Transactions on Networking*, vol. 21, no. 3, pp. 831–844, 2013.

[31] B. Cao, Y. Ge, C. W. Kim, G. Feng, H. P. Tan, and Y. Li, "An experimental study for inter-user interference mitigation in wireless body sensor networks," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3585–3595, 2013.

[32] X. L. Wang, Q. Gui, B. W. Liu, Z. P. Jin, and Y. Chen, "Enabling smart personalized healthcare: a hybrid mobile-cloud approach for ECG telemonitoring," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 3, pp. 739–745, 2014.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 418535, 19 pages
http://dx.doi.org/10.1155/2015/418535

Research Article

On Security Management: Improving Energy Efficiency, Decreasing Negative Environmental Impact, and Reducing Financial Costs for Data Centers

Katarzyna Mazur,[1] Bogdan Ksiezopolski,[1,2] and Adam Wierzbicki[2]

[1] *Institute of Computer Science, Maria Curie-Sklodowska University, Plac M. Curie-Sklodowskiej 5, 20-031 Lublin, Poland*
[2] *Polish-Japanese Academy of Information Technology, Koszykowa 86, 02-008 Warsaw, Poland*

Correspondence should be addressed to Bogdan Ksiezopolski; bogdan.ksiezopolski@acm.org

Received 26 March 2015; Revised 19 May 2015; Accepted 26 May 2015

Academic Editor: Bao Rong Chang

Security management is one of the most significant issues in nowadays data centers. Selection of appropriate security mechanisms and effective energy consumption management together with caring for the environment enforces a profound analysis of the considered system. In this paper, we propose a specialized decision support system with a multilevel, comprehensive analysis scheme. As a result of the extensive use of mathematical methods and statistics, guidelines and indicators returned by the proposed approach facilitate the decision-making process and conserve decision-maker's time and attention. In the paper we utilized proposed multilevel analysis scheme to manage security-based data flow in the example data center. Determining the most secure, energy-efficient, environmental friendly security mechanisms, we implemented the role-based access control method in Quality of Protection Modeling Language (QoP-ML) and evaluated its performance in terms of mentioned factors.

1. Introduction

The challenges faced by companies working in nowadays complex IT environments pose the need for comprehensive and dynamic systems to cope with the security requirements [1–3]. Security planning cannot answer all questions: we must take a step further and discuss a model for security management. One of the possible approaches to deal with this problem is to use the decision support system that is capable of supporting decision-making activities. Good decisions are usually result of detailed and precise examination of input parameters and comprehensive analysis of all the available alternatives. In order to solve a given problem, one should formulate it by using standardized methodologies and collect all the information that are relevant to the final decision. As the action of decision-making is usually a repeatable process, it is reasonable to embrace a much more comprehensive view of decision-making. In general, decision cycle consists of four main, ordered phases: *problem definition* (a decision situation that may deal with some difficulty or with an opportunity), *model construction* (in which one needs to describe the real world problem using specialized tools, designed specifically for this purpose), *identification and evaluation* (where the possible solutions to the modeled problems are identified and evaluated), and finally the *recommendation* and *implementation* stages, in which potential solutions are examined, compared, and chosen.

Since nowadays IT environments are combination of physical, virtual, and cloud infrastructures, legacy and modern technology components, they pose a need for going deeper, that is, performing more detailed, more complex, and integrated types of analysis. To meet this requirement, we propose the foundations of our decision support system for complex IT environments.

By definition, decision support system (DSS) is a conceptual framework which supports decision-making, usually by modeling problems and employing quantitative models for solution analysis. DSS is an interactive, programmed, computer-based system which helps decision makers to use available assets and resources, identify and solve problems,

complete decision process tasks, and make decisions. Being a multilevel, model-driven approach, our DSS lets one describe, examine, and analyse complex IT environments. Additionally, thanks to the extensive use of abstract models, it helps determine differences between distinct options. Such an approach is particularly helpful in the case of security-based data flow management, where one is capable of modeling miscellaneous scenarios (which differ in utilized security mechanisms) and assessing their quality. Managing interactions that occur across the complex secure systems characterized by the high level of dynamics, determining proper role definitions and permissions assignments, quantification of the authorization level of users allowed to perform various actions (based on the scope of the assigned role), is also crucial from the role-based access control point of view.

The main contributions of this paper are summarized as follows.

(1) We introduced the foundations of the comprehensive decision support system, together with presenting and describing its essential part, the multilevel analysis scheme utilized for the evaluation of complex secure systems.

(2) We extended previous studies on system's performance to a broader context: we discussed and examined new types of analyses, the financial and carbon dioxide emissions analyses (both being components of our new, utilized analysis scheme).

(3) We implemented proposed analysis techniques as modules utilized in Automated Quality of Protection Analysis (AQoPA) tool (which can be downloaded from the web page of the QoP-ML Project [4]); such solution gives the possibility to compare available security approaches in terms of time, energy, quality of protection, finance, and environmental impact.

(4) We illustrated the introduced approach by presenting the case study of the example data center with role-based access control implemented, since it has been widely adopted in secure IT environments and has been studied in many different contexts over the years (e.g., the economic [5] or security related studies [6]).

We organized the rest of this paper in the following way: in Section 2 we present the related work. Section 3 contains a brief overview of the Quality of Protection Modeling Language which we used in our research. Moving on to the next point, we demonstrate the foundations of the proposed decision support system and bring closer the multilevel analysis scheme. In Section 5, we introduce the CO_2 emission analysis phase, which tries to answer questions about the minimization of the ecological footprint of IT products and services. Moreover, we discuss the financial aspect of the data center performance and formalize its evaluation. To demonstrate the use of the proposed approach, in Section 6, we present the case study of the role-based access control as the example of the security-based data flow management. We prepare a RBAC model in Quality of Protection Modeling Language and assess its quality in terms of time, energy, quality of protection, finance, and environmental impact. Further, we discuss the results and finally summarize our work in Section 7.

2. Related Work

In the literature ([16, 17]) one can find miscellaneous types of decision support systems. Among them one can enumerate model-driven DSS [18], data-driven DSS, communication-driven DSS, document-driven DSS, and knowledge-driven DSS [19]. The model-driven decision support system approach is the most complex from the existing decision support system types. It deals with statistical, financial optimization or simulation models and uses input parameters and data provided by users to assist stakeholders in the decision-making process. In data-driven DSS, the data itself is the most significant part of the considered approach. Having easy access to a large amount of accurate, well-organized data sets stored in databases or data warehouses, the system seeks for specific information and reports retrieved data to users. With the rapid growth of the interconnected network environments, it became possible to utilize available network and communications technologies for the need of communication-driven decision support systems. Tools like groupware, video conferencing, and computer-based bulletin boards facilitate decision relevant collaboration and communication. Document-driven DSS is the most commonly used type of decision support system. It is focused on managing, retrieving, and manipulating unstructured information in a variety of electronic formats. To suggest possible solutions to a given problem, knowledge-based DSS makes use of expert systems and artificial intelligence. Simulating reasoning, explaining the logic behind its conclusion, knowledge-based DSS assists a decision-maker in an area where the specialized knowledge is required. The major goal of the security-driven data flow management is to simplify authorization management and review.

Having the ability of modeling various access control requirements and facilitating security administration process, RBAC became the object of the study of many researchers. In the literature [20, 21], one can find plenty of RBAC implementations. Preparing RBAC models in SecureUML [22] and UMLsec [23] authors usually focus on their economic or security aspects, omitting the influence of distinct authorization levels on system's efficiency. However, role-based access control has an undeniable impact on performance and should be implemented carefully in order to provide the required level of security together with energy efficiency and assurance of the security trade-offs. To address this issue, many modeling languages and tools have been proposed. Among them one can enumerate UMLsec and SecureUML presented by the researchers in [20]. Using the mentioned approaches, one is able to model and verify secure systems, either preexisting or those under construction. Nevertheless, introduced solutions focus on developing secure infrastructure or determining system's efficiency, rather than examining security and performance concerns at the same time. The traditional approach assumes that implementation of the strongest security mechanisms makes the system as secure as possible. Unfortunately, such reasoning can lead

Table 1: Established characteristics of QoP models.

	QA	E	Con	EE	H	Com	PE	FE	EA
Agarwal and Wang [7]	√	—	—	—	√	√	√	—	—
Ksiezopolski and Kotulski [8]	√	√	—	—	√	√	—	—	—
LeMay et al. [9]	—	√	√	—	—	—	—	—	—
Lindskog [10]	√	—	√	—	—	—	√	—	—
Luo et al. [11]	√	—	—	—	√	√	√	—	—
Ong et al. [12]	√	—	—	—	—	—	—	—	—
Petriu et al. [13]	—	√	√	—	—	√	√	—	—
Schneck and Schwan [14]	√	—	√	—	—	—	√	—	—
Sun and Kumar [15]	√	—	—	—	—	—	—	—	—
QoP-ML	√	√	√	√	√	√	√	√	√

to the overestimation of security measures, which causes an unreasonable increase in the system load [24, 25]. System's performance is especially important in environments with limited resources, including wireless sensor networks and mobile devices. Another example where such analysis should be performed is the cloud architecture. The latest research indicates three main barriers for using cloud computing: security, performance, and availability [26]. Unluckily, when the strongest security mechanisms are used, the system performance decreases, influencing its availability. This tendency is particularly noticeable in complex and distributed systems. The latest results show [24, 25] that in many cases the best way is to determine the required level of protection and then adjust security measures. (Among the means to meet these challenges one can indicate the security metrics [27].) Such approach is achieved by the means of the quality of protection models, where security measures are evaluated according to their influence on system's security.

Introduction of QoP term allows us to concentrate on security requirements in analysed system. In the literature, the security trade-offs are based on quality of protection (QoP) models. These models were created for different purposes and have miscellaneous features and limitations. In order to compare available security modelling approaches in terms of quality of protection, we prepared our set of qualities. Furthermore, we investigated different methodologies available in the literature and assessed them taking into account selected attributes. Proposed set of qualities (Table 1) along with their explanations and author's comments are presented below.

Approaches summarized in Table 1 can be characterized by the following main attributes.

(i) Quantitative assessment (QA) refers to the quantitative assessment of the estimated quality of protection of the system.

(ii) Executability (E) specifies the possibility of the implementation of an automated tool able to perform the QoP evaluation.

(iii) Consistency (Con) is the ability to model the system maintaining its states and communication steps consistency.

(iv) Performance evaluation (PE) gives the possibility of performance evaluation of the analysed system.

(v) Energy evaluation (EE) allows for the evaluation of the of energy efficiency of the analysed system.

(vi) Holistic approach gives the possibility of the evaluation of all security attributes.

(vii) Completeness is the possibility of the representation of all security mechanisms. This attribute is provided for all models.

(viii) Financial evaluation (FE) refers to the possibility of the financial assessment of the analysed system.

(ix) Ecological assessment (EA) answers questions about the environmental impact of the utilized mechanisms.

One can notice that only QoP-ML can be used for finding a trade-off between security (QA) and performance (PE) including energy efficiency evaluation (EE) of the system which is modeled in a formal way with communication steps consistency (Con). By means of QoP-ML, one can evaluate all security attributes (H) and abstract all security mechanisms which protect the system (C). Additionally, the QoP-ML approach is supported by the tool (E) required for the analysis of complex systems. It is also worth mentioning that only the QoP-ML allows for the financial assessment of the considered system (FE), giving the possibility of the evaluation of its environmental impact (EA).

According to the author's knowledge, Quality of Protection Modeling Language (QoP-ML), introduced in [28], is the only existing modeling language which satisfies all these requirements simultaneously. It allows for balancing security against the system efficiency, performing multilevel analysis, and extending the possibility of describing the state of the environment in detail. Quality of Protection Modeling Language permits determining the required quality of protection (QoP) and adjusting some of the security measures to these requirements, together with ensuring efficient system performance. This type of profound analysis can be accomplished by the help of the Automated Quality of Protection Analysis tool [4], which allows for the evaluation of the impact of every single operation defined in the prepared security model in terms of the overall system security. Additionally, in previous works, there were proposed and examined approaches which were successful also in assessing time, energy, and quality of protection of the analysed IT environments.

3. QoP-ML Overview

In the article [28] the Quality of Protection Modeling Language was introduced. Proposed solution provides the modeling language for making abstraction of cryptographic protocols that puts emphasis on the details concerning the quality of protection. The intended use of QoP-ML is to represent the series of steps, which are described as a cryptographic protocol. The QoP-ML introduced the multilevel protocol

analysis that extends the possibility of defining the state of the cryptographic protocol. Since approaches presented in the literature usually speak for an example of a model-driven security, in the light of the available development methodologies, QoP-ML excellently fits in a design known as a Model-Driven Engineering. The Model-Driven Engineering (simply known as MDE) is meant to focus on the creation and utilization of the abstract representations of the knowledge that govern a particular domain, rather than on the computing, algorithmic, or implementation concepts. Model-Driven Engineering approach is a broader concept than Model-Driven Architecture (MDA) or Model-Driven Security (MDS). MDE adds multiple modeling dimensions and the notion of a software engineering process. The various dimensions and their intersections together with a domain-specific language (DSL) form a powerful framework capable of describing engineering and maintenance processes by defining the order in which models should be produced and how they are transformed into each other. Serving as a domain-specific language, QoP-ML is capable of expressing security models in a formalized, consistent, and logical manner.

As is apparent from the above description, QoP-ML is a flexible, powerful approach to model complex IT environments. Therefore, we utilized it to prepare our case study and evaluate the quality of chosen security mechanisms using its supporting, automatic framework. In the following sections we present all the significant components of the language we utilized to create model for our scenario.

3.1. General Information. The structures used in the QoP-ML represent high level of abstraction which allows concentrating on the quality of protection analysis. The QoP-ML consists of processes, functions, message channels, variables, and QoP metrics. Processes are global objects grouped into the main process, which represents the single computer (host). The process specifies behavior, functions represent a single operation or a group of operations, and channels outline the environment in which the process is executed. The QoP metrics define the influence of functions and channels on the quality of protection. In the article [28] the syntax, semantics, and algorithms of the QoP-ML are presented in detail.

3.2. Data Types. In the QoP-ML, an infinite set of variables is used for describing communication channels, processes, and functions. The variables are used to store information about the system or specific process. The QoP-ML is an abstract modeling language, so there are no special data types, sizes, or value ranges. The variables do not have to be declared before they are used. They are automatically declared when used for the first time. The scope of the variables declared inside the high hierarchy process (`host`) is global for all processes defined inside `host`.

3.3. Functions. The system behavior is changed by the functions, which modify the states of the variables and pass the objects by communication channels. During the function definition, one has to set the arguments of this function which describe two types of factors. The functional parameters, which are written in round brackets, are necessary for the execution of the function and the additional parameters which are written in square brackets and have an influence on the system's quality of protection. The names of the arguments are unrestricted.

3.4. Equation Rules. Equation rules play an important role in the quality of protection protocol analysis. Equation rules for a specific protocol consist of a set of equations asserting the equality of function calls. For instance, the decryption of the encrypted data with the same key is equal to the encrypted data.

3.5. Process Types. The processes are the main objects in the QoP-ML. The elements which describe the system behavior (functions, message passing) are grouped into processes. In the real system, the processes are executed and maintained by a single computer. In the QoP-ML the sets of processes are grouped into the higher hierarchy process named `host`. All of the variables used in the high hierarchy process (`host`) have a global scope for all processes which are grouped by the host. Normally, the variables used in the `host` process cannot be applied to the other high hierarchy process. This operation is possible only when the variable is sent by the communication channel.

3.6. Message Passing. The communication between processes is modeled by means of channels. Any type of data can be passed through the channels. The channels must be declared before the data is passed through. The data can be sent or received by the channels. The channels pass the message in the FIFO order. When the channels are declared with the nonzero buffer size, the communication is asynchronous. The buffer size equal to zero stands for the synchronous communication. In synchronous communication, the sender transmits the data through the synchronous channel only if the receiver listens to this channel. When the size of the buffer channel equals at least 1, then the message can be sent through this channel even if no one is listening to this channel. This message will be transmitted to the receiver when the listening process in this channel is executed.

3.7. Security Metrics. The system behavior, which is formally described by the cryptographic protocol, can be modeled by the proposed QoP-ML. One of the main aims of this language is to abstract the quality of protection of a particular version of the analysed cryptographic protocol. In the QoP-ML, the influence of the system protection is represented by the means of functions. During the function declaration, the quality of protection parameters is defined and details about this function are described. These factors do not influence the flow of the protocol, but they are crucial for the quality of protection analysis. During that analysis, the function's quality of protection (QoP) parameters are combined with the next structure of QoP-ML, the security metrics. In this structure, one can abstract the functions' time performance,

their influence on the security attributes required for the cryptographic protocol, or other important factors during the QoP analysis.

4. Model-Based Decision Support System

In this section we introduce the foundations of our approach to the creation of a model-based decision support system. To perform a detailed, profound analysis of secure systems, focusing on every aspect of the examined environment, many different components need to be taken into consideration. To facilitate this complex process in interrelated IT systems, we designed, implemented, and utilized the model-based decision support system. Introduced DSS helps in the evaluation of consequences of given decisions and may advise what decision would be the best for achieving given set of goals. Being a well-organized, consistent solution, the proposed approach allows for a precise, specific analysis of the studied systems, making the analysis process multilevel (regarding time, energy, financial costs, environmental impact, and quality of protection). In following sections all stages of the proposed DSS are presented.

Step 1 (problem definition). The formulation of the problem may be defined as the process of acquiring and organizing knowledge in any situation on which the decision-maker intends to act. In this stage potential problems are identified and described. To protect the confidentiality, integrity, and/or availability of the considered system, security objectives need to be determined as early in this phase as possible. This can be done by following best practice recommendations on information security management defined in ISO/IEC 27002 standard [29]. Through the standardized approach to security objectives definition, we can assume that they reflect the high-level goals of the system in providing an appropriate secure environment for users of the system.

Step 2 (model construction). To organize and examine existing alternatives accurately, the modeling process should be appointed. The quality of the decision relies upon the correctness and accuracy of the modeling solution. Model creation aims at creating model of the chosen IT environment with the use of the Quality of Protection Modeling Language. QoP-ML, as a dedicated, specialized solution, uses qualitative models that help highlight important connections in real world systems and processes and recommend the solution to a given problem.

Security Metrics Definition. To be able to make a good decision, one needs to have relevant and accurate information, which help in choosing the best solution from existing alternatives. Right decisions impose a requirement for input parameters to be as solid and adequate as possible. When the model is ready, there comes the time when its input data should be gathered. Obtaining robust, repeatable security metrics, one should rely on the results gathered by Crypto Metrics Tool, since the framework was designed to use statistics to ensure most reliable measurements. Robust method, proposed by the authors and utilized further in our case study, involves characterizing a system based upon its statistical parameters. The approach examined and discussed in [30] guarantees that obtained results are accurate and free of random errors. The proposed tool yields the results in a form that is appropriate for the Automated Quality of Protection Analysis tool. Crypto Metrics Tool can be downloaded from the webpage of the QoP-ML project [4].

Scenarios Definition. Scenarios represent different versions of the evaluated model. They are also known as versions, in such one can assess the quality of protection of the modeled environment, using miscellaneous security mechanisms and applying them to the same model.

Data Flow Management. In any e-business solution, it is integral to manage every piece of data. To provide high quality services and drive successful business, company must have complete, accurate, combined data available in a timely manner. It is relevant to think about data flow as it pertains to every functional requirement, what kind of data with what kind of system within what time frame, in order to maintain efficiency and control throughout every process. To meet the given requirements, we proposed adding a new stage to the analysis process, the data flow management. Introduced phase uses a data flow management to help reduce the security risks, preventing security vulnerabilities at the same time. The main goal of this stage is to indicate most suitable mechanisms that optimize the data flow and make data available as quickly and consistently as possible.

Step 3 (identification and evaluation). This stage consists of multimeasure cross-validation and assessment of the modeled alternatives. Since QoP-ML provides the possibility of multilevel analysis, in this paper, we decided to extend this phase and divide it into five connected substeps. Time analysis, energy analysis, and QoP analysis [31] were proposed in previous works and implemented in Automated Quality of Protection Analysis tool. In this paper, we aim to introduce additional types of analysis: the finance estimation and carbon dioxide emissions evaluation.

Below we outline the underlying concepts of each of the proposed analysis phases to utilize them further in the paper. As shown in Figure 1, the time analysis is the analysis on which three remaining analyses are based. Therefore, before we go any further in introducing our new proposals, let us focus on time, energy, and quality of protection analyses.

The Time Analysis. The aim of the time analysis is to estimate the time, which was taken by all the security operations performed during the execution of the cryptographic protocol. The time analysis helps to determine mechanisms which are the most time efficient between the proposed security operations: it can be also useful when determining the total number of users that can be handled by the server within a given time interval.

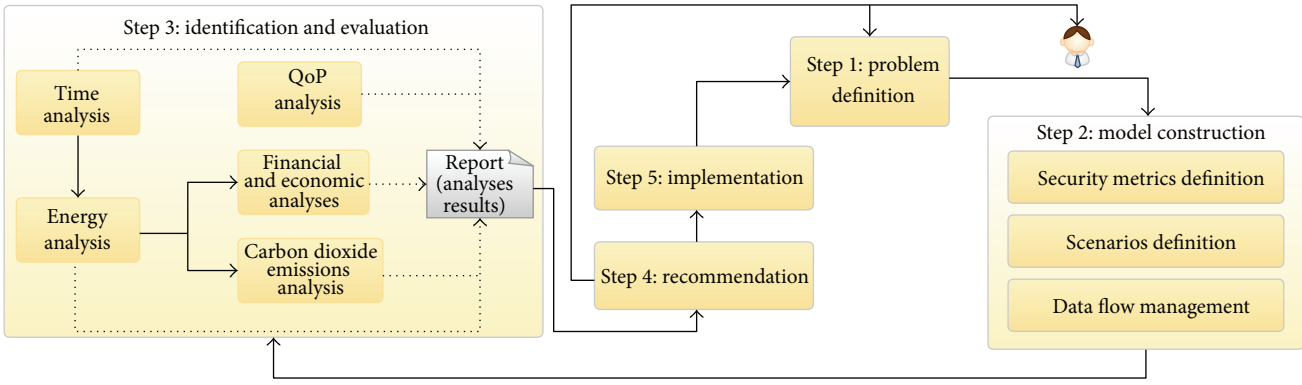


Figure 1: Utilized decision support system.

The Energy Usage Analysis. Besides the time analysis, we proposed including the power consumption to the analysis process, in order to evaluate the energy consumption of the modeled system. To obtain the total amount of the energy consumed by the security operations, the time analysis module must be included in the performance analysis process, since it tracks the time of operations and communication steps. The energy consumption is calculated as the sum of energy consumed by operations that use CPU. (The details can be found in [32].)

The QoP Analysis. Another crucial aspect of the multilevel analysis is the assessment of the security quality. Such analysis is based on the evaluation of the impact of security mechanisms on system performance. Estimation of quality of protection (QoP) is a challenging task: utilized approach should be flexible enough to allow evaluating quality of protection of different versions of cryptographic protocols (security policies) in an economic manner. In [31] the authors introduced the framework which permits assessing the quality of protection of previously predefined security mechanisms and for not directly defined security configurations. Proposed solution, being an automated approach, lets one define all possible scenarios for all IT processes, which can be very complex and in many cases not feasible with other existing frameworks.

New Type of Analyses: The Financial and Economic Analyses. The reason for the introduction of the new stage to the utilized analysis scheme is the crucial role of finances in IT. As data center is one of the most financially concentrated assets of any organization, there is a high demand for a standardized method for measuring the total cash outlay of the physical infrastructure of data centers. Designing IT budgets, identifying IT budgets items, developing appropriate pricing strategies, and implementing and operating financial management enable companies to get more purchasing power out of their budgets and preserve cash for operational issues. One of the essential parts of the enterprise economic policy is the effective management of the utilized power, which is as well covered in our pleasantly new analysis scheme. In large IT environments, consisting of tens, if not hundreds, of thousands of working machines, electricity is the factor which generates one of the biggest expenditures.

New Type of Analyses: The Carbon Dioxide Emissions Analysis. The recent growth of data centers requires more energy-efficient server design. However, these days saving some money on energy bills is one thing, but reducing the CO_2 is a much more admirable goal. Since data centers are a large fraction of the IT, there is a high demand for lowering emissions of CO_2 (and, in turn, bills) by reducing power consumption. Besides servers, data centers also use storage and network components, which as well produce huge amounts of carbon dioxide. Estimating the amount of the CO_2 emissions, it is important to note that its total quantity depends on miscellaneous factors, such as the size of the data center (number of all its working components), server load (which translates into the utilized kilowatt-hours), and the type of resource utilized to generate electricity. Hence, machines, which consume a great deal of power, cause a negative environmental impact. As stated before, reduction of the energy usage from green computing techniques translates into lower emissions of carbon dioxide, stemming from a reduction in the resources used in power plants.

Steps 4 and 5 (recommendation and implementation). After the choice phase all alternatives are searched and evaluated and one of them is chosen as recommended solution. The chosen decision is to be carried out. Only if the recommended solution is successfully implemented, the problem is considered solved.

5. The Financial, Economic, and Carbon Dioxide Emissions Analyses

In this section we present and briefly describe our two new modules utilized in the proposed decision support system. Introduced equations are used to calculate the financial aspect of the data center maintenance. Further in this section we discuss the environmental impact of the data center management. By means of the proposed method and corresponding formula, we present a simple approach for

estimating the rough amount of the carbon dioxide released to the atmosphere.

5.1. The Financial Analysis. Calculating total operating cost of a data center, one needs to take into account both *fixed* and *variable* costs, which are affected by complex and interrelated factors. In this paper, performing the finance analysis, we took into consideration only costs of power delivery and cooling infrastructure utilization, as they refer to the *variable* expenditures. Both power delivery and cooling costs respond directly and proportionately to changes in input parameters (such as utilized security mechanisms, scenarios, and metrics). Any change in any of the input parameters can have the influence on those expenditures. When it comes to the *fixed* costs, (such as hardware amortization expenditures, hardware and software purchase costs, and personnel salaries), we simply omitted them in our analysis, since they are independent, remain more or less unchanged irrespective of the input parameters, remain constant throughout the relevant range, and are usually considered sunk for the relevant range (not relevant to output decisions).

Below we introduce general formulas used for calculating total energy and cooling costs.

5.1.1. Cost of Power Delivery. The design of the electrical power system must ensure that adequate, high quality power is provided to each machine at all times. One should pay special attention to the relationship between the CPU utilization and energy consumed by other components of a working server. Since the amount of the power consumed by a single machine consists of the energy usage of all its elements, it is obvious that a decrease in the energy consumed by the CPU will result in a lower energy consumption in total. Thus, total cost of both power delivery and utilization can be summarized as follows:

$$\varsigma_{\text{power}} = \kappa_{\text{busy+idle}} \cdot \sigma \cdot \chi \cdot \rho, \tag{1}$$

where $\kappa_{\text{busy+idle}}$ is the total amount of the utilized kilowatt-hours by the server, σ is the cost of a one kWh, χ is the total amount of hours when the server was busy, and ρ is the total amount of days when the server was busy.

However, total power consumption of a single server is a sum of the power utilized by all its working components. In such case, $\kappa_{\text{busy+idle}} = \kappa_{\text{CPU}} + \kappa_{\text{RAM}} + \kappa_{\text{HDD}} + \cdots$. Therefore, to assess the real cost of a single working machine, the approximate amount of the energy utilized by all its elements should be estimated.

In general, electricity provided by the electric company and consumed by its customers is measured in kilowatt-hours. Being aware of the price of one kWh and knowing that CPU worked χ hours through ρ days, utilizing κ kilowatt-hours, it is fairly straightforward to calculate the total financial cost of its work, using the formula analogous to (1). Before we start further evaluation of the energy consumed by the CPU, we need to make some assumptions about its utilization. Let us introduce the simplified CPU utilization formula:

$$U = \frac{R}{C}, \tag{2}$$

where U is the CPU utilization, expressed in percentage, R defines our requirements, the actual busy time of the CPU (seconds), and C stands for the CPU capacity, the total time spent on analysis (seconds).

Usually, the CPU utilization is measured in percentage. In the formula introduced above, R refers to the time we require from the CPU to perform an action. This time is also known as the *busy* time. CPU capacity can be expressed as the sum of the *busy* and *idle* time (i.e., the *total* time available for the CPU). Going simple, one can say that over a 1-minute interval, the CPU can provide a maximum of 60 of its seconds (power). The CPU *capacity* can then be understood as *busy time* + *idle time* (the time which was used plus the one which was left over). Using the above simplifications, when going multicore, CPU capacity should be multiplied by the number of the CPU cores ($C = C \cdot \text{cores}$). In context of served requests, presented equation (2) can be further detailed as follows:

$$\text{load}\,[\%] = \frac{\text{time}_{\text{session}} \cdot \text{users}}{\text{time}_{\text{total}}}, \tag{3}$$

where $\text{time}_{\text{session}}$ refers to the time the single request took (seconds), users stands for the number of incoming user connections to be managed, and $\text{time}_{\text{total}}$ is expressed as $\text{time}_{\text{session}} \cdot \text{users} + \text{time}_{\text{idle}}$ and represents the total time taken by all the handled connections together with the one which was left over.

5.1.2. Cost of Cooling Infrastructure Utilization. Providing sufficient cooling is essential to ensure reliable running of servers, routers, switches, and other key data center pieces of equipment. As the cooling infrastructure absorbs energy to fulfill its function, the cost of cooling needs to be included in the total cost of the server maintenance. To keep servers operational, cooling a server consumes defined amount of watts for every watt that powers it, depending on cooling efficiency. Same as in the case of power delivery, to keep server rooms temperature within the listed tolerances, there is a requirement for additional, back-up cooling solutions. Back-up chillers generate additional costs, which must be taken into account as well. To obtain an approximate amount of the power consumed by the cooling, one can use the equipment heat dissipation specifications, most often expressed in British Thermal Units (BTUs), generally available either in the system users guide or on the manufacturers website. Specifications state how many BTUs are generated in each hour by the individual machine. Therefore, the formula for calculating the cooling cost to keep the equipment in normal operating conditions is given as follows:

$$\varsigma_{\text{cooling}} = \text{BTU}_{\text{cooling}} \cdot \sigma \cdot \chi \cdot \rho, \tag{4}$$

where $\text{BTU}_{\text{cooling}}$ is the amount of the BTUs generated by the cooling system (per server), σ is the cost of a one kWh, χ is

the total amount of hours when the server was busy, and ρ is the total amount of days when the server was busy.

Knowing that one watt is equal to 3.412 BTU/hour (e.g., using 100 watts of power generates 341.2 BTU per hour), the above formula becomes analogous to (1).

5.1.3. Total Cost. Key elements in data center budgets are the power delivery system, the networking equipment, and the cooling infrastructure. Besides the above most-crucial factors, there exist additional costs associated with data center operation, such as personnel and software expenses. Therefore, the real operating cost of the data center can be expressed as

$$\varsigma_{\text{total}} = \varsigma_{\text{power}} + \varsigma_{\text{cooling}}. \quad (5)$$

As mentioned before, in our case we simply exclude *fixed* costs from our analysis, since they do not vary with changes in input parameters.

5.2. The Carbon Dioxide Emissions Analysis. Using simple formula one can easily estimate the annual environmental impact (ς_{CO_2}) for the considered CPU (and, analogously, for the single server, server room, and, of course, the whole data center):

$$\varsigma_{CO_2} = \kappa \cdot \chi \cdot \rho \cdot \delta, \quad (6)$$

where κ is the total amount of the utilized kilowatt-hours, χ is the total amount of hours when the server was busy, ρ is the total amount of days when the server was busy, and δ defines the amount of pounds of CO_2 per kWh.

The amount of carbon dioxide (CO_2) produced per kilowatt-hour (kWh) depends on the type of the fuel utilized to generate electricity. One can calculate the amount of carbon dioxide produced per kilowatt-hour for specific fuels and specific types of generators by multiplying the CO_2 emissions factor for the fuel (in pounds of CO_2 per million BTU) by the heat rate of a generator (in BTU per kWh generated) and dividing the result by 1 000 000.

6. Case Study: Security-Based Data Flow Management in Data Center

In the paper we utilized the case study approach to obtain profound, in-depth understanding of the introduced, complex, multilevel analysis method in its potential implementation context, which is, in our case, the security-based data flow management, described in detail in latter sections. To demonstrate the use of the presented system and its analysis scheme, we proposed using the role-based access control approach, prepared a scenario, and assessed its quality. We made use of QoP-ML framework and created by its means the role-based access control model (since it is an excellent example of the data flow management), to examine the quality of chosen security mechanisms in terms of time, energy, quality of protection, finance, and environmental impact. Below we use the model-based decision support system proposed in Figure 1 and briefly describe its steps in following subsections.

6.1. Example DSS Implementation

Step 1 (problem definition). The decision-making process begins when one identifies the real problem. In our case, we managed to formulate questions about the fastest, the most energy-saving, the most secure, the cheapest, or the most green security mechanism among available solutions, on the example of the role-based access control model.

Addressing the security objectives issue, we provided a clear definition of what the system should do to counter and manage threats to its security. Our goal was to examine how to achieve a balance between the performance and security. We managed to perform such analysis by striving to accomplish example security objectives, created on the basis of guidelines from ISO/IEC 27002:2013, using miscellaneous security mechanisms. Table 2 contains established security objectives together with their example realization in our model.

Step 2 (model construction). Before we adopt QoP-ML approach to our needs, utilize analysis scheme, and finally perform the actual estimation of the daily, weekly, monthly, and annual server impact in terms of money and environmental effects, let us give some assumptions about the potential implementation environment (Figure 2). Consider a call center company located in Nevada, USA, managing a typical IT environment of 42U server racks (520 physical servers in total, 13 physical servers per rack). Suppose that the example enterprise uses electricity provided by the electric company, which measures energy consumption in kilowatt-hours. Examined enterprises have many departments with miscellaneous responsibilities, and thus distinct permissions and rights to the company's assets. Given a specified load capacity, servers handle enterprise's traffic continuously for 24 hours. In the example enterprise, we have 1 000 CSRs' workstations (which automatically translates into the number of the users assigned the first role), 10 security managers (employees having role's 3 permissions), and 20 system operators, people with a second level of authorization. Each server in the example call center is equipped with the Intel Xeon X5675 processor, being able to handle the required number of employees' connections, regardless of the assigned RBAC role. To simplify, in our analysis, we assumed that on average single Dell PowerEdge R710 server consumes about 0.3 kWs per hour on performing its daily routine tasks and added the amount of power consumed by the CPU in each role.

After a brief introduction of the potential implementation environment, let us now move on to the example utilization of our novel analysis scheme, where we, step by step, examine the system in detail.

We proposed preparing the role-based access control in the Quality of Protection Modeling Language to improve security management, thereby enhancing the security itself, and analyse security-based data flow management impact on the whole system performance in the context of time and energy consumption, quality of protection, cost savings, and environmental impact.

Table 2: Security objectives established according to the guidelines provided by ISO/IEC 27002:2013.

Security objective	ISO/IEC 27002:2013 section	Realization
(1) Security management and access control (SO1)	5.1.1b: assignment of general and specific responsibilities for information security management to defined roles 6.1.1c: authorization levels should be defined and documented 9.1.2c: authorization procedures for determining who is allowed to access which networks and networked services 9.1.2d: the means used to access networks and network services [should be defined] 9.1.2e: user authentication requirements for accessing various network services 9.2.3a: the privileged access rights associated with each system or process and the users to whom they need to be allocated should be identified	Data flow management: RBAC role assignment
(2) Confidentiality, authenticity, or integrity protection (SO2)	10.1.1a: the management approach towards the use of cryptographic controls across the organization 10.1.1b: the required level of protection should be identified taking into account the type, strength, and quality of the encryption algorithm required 10.1.1c: the use of encryption for protection of information transported across communication lines	Communication channel protection (TLS)
(3) System capacity management (SO3)	12.1.3: the use of resources should be monitored and tuned and projections made of future capacity requirements to ensure the required system performance	Monitoring and analysis of server resources with distinct security mechanisms applied: the multilevel analysis process
(4) Assurance of correct and secure operation of information processing and handling facilities (SO4)	12.1.1b: procedures should specify the operational instructions for processing and handling of information both automated and manual	Secure access to FTP, web, . . ., and servers
(5) Maintenance of the integrity and availability of information (SO5)	12.3.1a: accurate and complete records of the back-up copies and documented restoration procedures should be produced 12.3.1c: the back-ups should be stored in a remote location, at a sufficient distance to escape any damage from a disaster at the main site	Information back-up: secure access to data storage
(6) Protection of information involved in electronic messaging (SO6)	13.2.3a: protecting messages from unauthorized access, modification, or denial of service 13.2.3f: stronger levels of authentication controlling access from publicly accessible networks	Secure access to e-mail server

In complex secure environments, aside from enhanced security and reduced administration costs, system performance is another important artifact that needs to be carefully evaluated. Modeling a complicated enterprise infrastructure and applying the role-based access control is a challenging task. Instead of introducing a complex infrastructure abstraction and considering all the possible operations, we managed to model a simple RBAC usage in an example business situation (i.e., we examined communication steps that occur in a single client accessing a single server) and evaluated its performance with the use of the Automated Quality of Protection Analysis tool. Due to the page limitation, complete RBAC model cannot be presented in the paper. However, it can be downloaded (along with the Automated Quality of Protection Analysis tool) from the web page of the QoP-ML project [4].

Modeling the RBAC approach, we defined QoP-MLs functions, equations, channels, processes, subprocesses, and hosts. Let us now briefly discuss utilized QoP-ML structures we prepared to create the role-based access control model. To create the role-based access control in the Quality of Protection Modeling Language, we prepared a security model consisting of two communicating hosts: a client and a server. In addition, we prepared three asynchronous communication channels to facilitate the information exchange process. On the client's site, we modeled the main process being responsible for establishing secure connection with the server and a subprocess capable of generating different types of network traffic based on the role received from the server. Server abstracted in Quality of Protection Modeling Language is much alike the client; it also has a main process which sets up the communication parameters, but, opposite to the client, it contains three subprocesses, thereby being able to manage clients with miscellaneous levels of authorization. Modeling the RBAC, we defined QoP-MLs functions, equations, channels, processes, and hosts. In this section we present and discuss only the essential QoP-ML elements we prepared to create the security model.

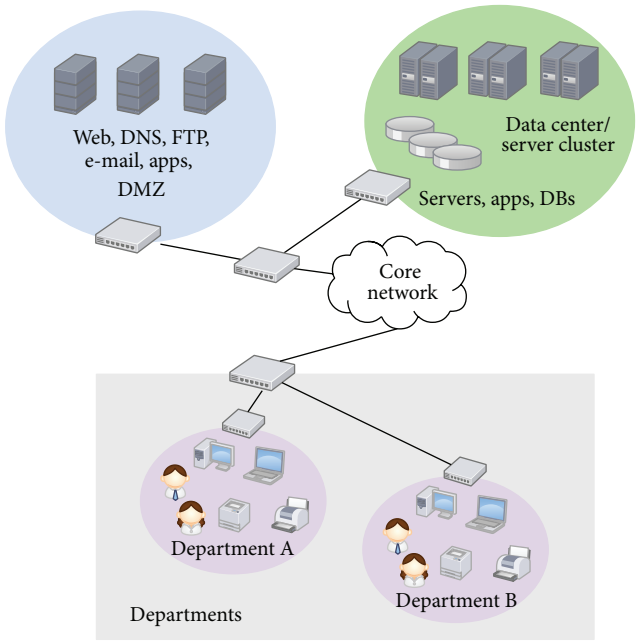


Figure 2: Example enterprise network architecture.

Functions defined in Quality of Protection Modeling Language refer to the roles specified for the example enterprise. Declared operations represent 3 roles: *role 1*, *role 2*, and *role 3*. Along with *functions* we declared some *equational rules*.

Since one needs communication between running processes, it is necessary to define QoP-MLs channels. Defined channels are used to exchange the TLS traffic, actual data traffic, and the assigned RBAC role.

In our approach, subprocesses express operations that may be performed by users with different RBAC roles assigned. Client and server processes are used to model the TLS handshake operations. After defining processes and subprocesses, one can group them into `host` structures, named *client* and *server*, which express the communicating sites in the RBAC model. An example client's subprocess connects to the FTP server, while server's subprocess handles the request, according to the assigned role. Notice that the server handles distinct roles differently.

Security Metrics Definition. It is worth noticing that QoP-ML provides the ability to determine modeled system performance on machines with different hardware specifications. To examine the hardware impact on access control management, one can use QoP-ML's feature, the security metrics, described earlier in this paper. We collected data required by the *Security Metrics Definition* stage using the Crypto Metrics Tool and performing the metrics generation process on a server equipped with the Intel Xeon X5675 processor. Using CMTool we gathered security metrics for all the security mechanisms proposed in our case study (Box 1). We took into consideration just one of the available base measures: the CPU time. In our case, the CPU time indicates the amount of time for which a central processing unit was used for processing, the execution of the security operations (such as AES encryption/decryption, SHA-1 hashing, and others available). Obtained, free of random errors values (along with their attributes) are gathered in Table 3.

TLS version 1
(security level = *low*)
RC4 + MD5
TLS version 2
(security level = *medium*)
AES/CBC 128 + SHA-1
TLS version 3
(security level = *high*)
AES/CBC 256 + SHA-512

Box 1: TLS protocol versions together with corresponding cryptographic algorithms.

Remaining characteristics needed for the analysis (such as the CPU voltage) were taken from the official documentation of the processor which was utilized for the metrics generation [33]. Since cryptographic operations utilized in the defined scenario are considered power-consuming, in our analysis, when evaluating energy utilized in the *busy* state of CPU, we decided to choose its maximum available voltage. Assessing the power consumed in the *idle* state of the CPU, one should assume the smallest available voltage value. All the utilized values are gathered in Table 4.

Scenarios Definition. To emphasize and prove role's influence on data flow management, system performance, and adequate security objectives realization, we prepared and analysed a simple scenario. Prepared scenario refers to the example business situation and possible role assignment in the actual enterprise environment. Given the enterprise network infrastructure in Figure 2, consider having three roles: *role 1*, *role 2*, and *role 3* with corresponding security levels: *low*, *medium*, and *high*. The role's permissions are defined as follows.

(i) *Role 1*. Users assigned to this role have the access to the e-mail, FTP, web, and application servers with the communication channel protected by means of the TLS protocol in version 1; moreover, they are allowed to access data centers in the enterprise's DMZ.

(ii) *Role 2*. This represents users with greater responsibilities and thus stronger protection mechanisms of the actions they are allowed to perform. Being a member of the role 2, user is able to access e-mail, FTP, web, and application servers with a communication channel secured by the TLS version 2. Furthermore, members of the *role 2* are permitted to access servers $S1$ and $S2$ in the production DMZ.

(iii) *Role 3*. Members of the role 3 are the most authorized users among considered roles. They are allowed to access e-mail, FTP, web, and application servers with the communication channel protected by means of the TLS protocol in version 3, permitted to perform actions on production management resources and on general office assets, and process data available within

Table 3: Security metrics gathered by Crypto Metrics Tool for the purpose of our case study.

Security mechanism	Characteristics			
	Operation type	Operation mode	Key length	Base measure (CPU time)
AES	Encryption Decryption	CBC	128 bytes	0.0000043972 ms
AES	Encryption Decryption	CBC	256 bytes	0.0000060001 ms
RSA	Encryption Decryption	—	2048 bytes	0.0002539800 ms
RC4	Encryption Decryption	Stream	128 bytes	0.0000012571 ms
SHA-1	hmac	—	—	0.0000016390 ms
SHA-512	hmac	—	—	0.0000028952 ms
MD5	hmac	—	—	0.0000014137 ms

Table 4: CPU specific values used for our estimation.

Power consumption	$\text{Voltage}_{\text{min}}$	$\text{Voltage}_{\text{max}}$	$\text{Current}_{\text{busy}}$	$\text{Current}_{\text{idle}}$
95 W	0.75 V	1.35 V	W/V ≈ 70.37 A	W/V ≈ 126.67 A

DMZ data centers with the strongest communication protection mechanism.

Using described roles we managed to build different versions which comprise example implementation scenario, summarized in Table 5.

As it is clear from Table 5 and Box 1, different roles use distinct security mechanisms. *Role 1* makes use of two cryptographic algorithms, namely, RC4 and MD5. RC4 is the most widely used software stream cipher. The cipher is fairly simplistic when compared to competing algorithms of the same strength and boasts one of the fastest speeds of the same family of algorithms. It should be chosen when the performance is the main concern.

In TLSv2 we utilized AES in CBC mode with 128-bit key and SHA-1. AES is a symmetric-key algorithm which makes use of the same key for both decrypting and encrypting information. AES is not great in that its strength comes through excessive CPU effort but it is of considerably greater strength (both theoretically in real world attacks) than RC4.

Last but not least, the *most secure* role, *role 3*, utilizes for its functioning AES in CBC mode with 256-bit key and SHA-512 as a cryptographic hash function. If the performance is the main concern, one should choose AES-128, since AES-256 is not much more secure in practice (128-bit keys cannot be brute forced anywhere in the foreseeable future).

Data Flow Management. To examine the performance of miscellaneous roles in a real-life situation, we mapped the general rules proposed in the defined scenarios to the example segment of the enterprise network in a call center company. Considering the extended example, we can actually prove that the chosen role matters if it comes to the system performance and the energy usage. In our case study, users assigned to the given role have different responsibilities they perform. The role of the user is usually determined by her/his responsibilities within the company. Such extension clarifies our approach: the need for the RBAC and different security levels is undeniable. All network traffic in the enterprise needs to be secure, but users who manage the most valuable enterprise assets should use the most secure network connections, since any security vulnerability or weakness can compromise the integrity, availability, or confidentiality of the enterprise crucial resources and thus expose the enterprise to serious costs.

In that case, we mapped the third role to the category of users called *security manager*, since the mentioned role has the highest privilege in the system managing crucial assets and thus needs the highest possible security level to ensure required security. Users assigned the mentioned role have access rights that enable them to perform management level operations. Their permissions do not allow them to perform application related activities such as usage of the Customer Service's VoIP software. They perform the following, example operations: User Profile Maintenance, Task Access Control, System Security, and, the most important from the company's point of view, Communication System Maintenance (VoIP Software Maintenance). Call centers function efficiently when they have strong communications system consisting of voice over IP technology or VoIP. With a reliable VoIP software technology, a call center would be able to render outstanding communication service with the least risk of downtime occurrence.

The second role introduced in our scenario is equivalent to the permissions of the *system operator* in the example enterprise network. The activities of the users assigned the *system operator* role are as follows (but not limited to): File Transfer (FTP), Archival/Retrieval, and the rest. However, *system operators* are not allowed to perform any application or security manager related activities. Since *system operators* have access to the less significant enterprise assets, the security level of the role can be, respectively, lower.

Table 5: Scenario defined for our case study.

	Scenario		
	Role 1	*Role 2*	*Role 3*
Access type	E-mail, FTP, web, applications, data center servers	E-mail, FTP, web, applications, server S1 in DMZ, server S2 in DMZ	E-mail, FTP, web, applications, data center servers, DMZ production servers, General office
Data size (for each action separately)	E-mail: 8 MB FTP: 10 MB Web: 3 MB Applications: 4 MB Data center: 25 MB	E-mail: 10 MB FTP: 10 MB Web: 4 MB Applications: 6 MB Server S1: 5 MB Server S2: 15 MB	E-mail: 10 MB FTP: 5 MB Web: 4 MB Applications: 6 MB Data center: 15 MB DMZ production servers: 8 MB General office: 2 MB
Security mechanisms	TLSv1 (see Box 1)	TLSv2 (see Box 1)	TLSv3 (see Box 1)
Security level	*Low*	*Medium*	*High*
Security objectives	1, 2, 3, 4, 5, 6	1, 2, 3, 4, 5, 6	1, 2, 3, 4, 5, 6

The role which uses the *role 1* security parameters is assigned to the ordinary users of the system, namely, employees, the Customer Service Reps (CSRs), who have access to the enterprise applications and server's resources. They cannot perform any security manager or system operator activities. Their role within the system includes the usage of the enterprise-specific applications (the VoIP software, self-services software, speech technology applications, and many more) provided by the enterprise servers (being under the supervision of the security managers).

Users assigned the defined roles can use the same types of enterprise resources (FTP, WWW, applications, databases, and others too numerous to mention). However, in fact, they work with different physical assets. As mentioned above, single session performed by the user consists of operations appropriate for the user's role.

Step 3 (identification and evaluation). During our research, we were able to perform the actual analyses for only one client accessing the server in a single session. Remaining results were evaluated to grow linearly, along with the number of incoming session requests. We examined time spent on handling client connections, accompanying amount of the consumed kilowatt-hours, and the resulting financial costs along with emissions of carbon dioxide.

To satisfy the *confidentiality, authenticity*, *or integrity protection* security objective we assumed that all the utilized applications are tunnelled by miscellaneous TLS protocol versions. Proposed variants of the TLS protocol together with equivalent cryptographic algorithms are summarized in Box 1.

In following subsections we present and discuss obtained results and consider remaining security objectives. In time, energy, and QoP analyses, we estimated the results that can be applied to the working central processing unit. Broader picture is presented in financial, economic, and environmental analyses, where we took into consideration not only the CPU but the server itself.

The Time Analysis. Consider, for instance, *role 1*, in the first scenario. Here, the user has access to the e-mail, FTP, WWW, and application servers and to the data center resources. A single session between the client and the server can carry the traffic with the maximum size of 50 MB. The communication channel is protected by the TLS protocol in version 1 and it takes exactly 0.28 seconds to perform (Table 6). Nevertheless, having identical conditions, changing only the channel protection type, time extends to 1.2025 seconds (for *role 2*). As *role 3* is the most secure communication type example (having other conditions equal to *role 1* and *role 2* at the same time), it is reasonable to presume that it takes the longest time to accomplish (1.2683 seconds). Such analysis provides serious argumentation to believe that the assigned role (thus the level of protection) can influence data flow and thus overall system performance.

Supposing that the time of the chosen security operation assessed by the time analysis module was equal to τ seconds and assuming that there was a given time interval equal to ϕ = 3 600 seconds, for the server working under φ% of machine load, it is quite straightforward to calculate the maximum number of users the server is able to manage within the given time interval:

$$\text{users}_{\max} = \frac{\varphi \cdot \phi}{100 \cdot \tau}, \tag{7}$$

where τ is the total time of a single session in seconds, φ is the machine load in percentage, and ϕ is the considered time interval in seconds.

However, to prove our hypothesis, we estimated the hourly server load of the server being accessed by users with distinct roles. Utilizing results obtained by AQoPA along with

Table 6: Server's performance results obtained by AQoPA suggest that the assigned role (and thus, the proper data flow management) matters if it comes to the system's performance.

Action performed (access)	Scenario RBAC role		
	Role 1	*Role 2*	*Role 3*
E-mail	0.0448 s	0.1266 s	0.1865 s
FTP	0.0560 s	0.1266 s	0.0932 s
Web (WWW)	0.0168 s	0.0506 s	0.0746 s
Application(s)	0.0224 s	0.0760 s	0.1119 s
Data center	0.1400 s	0.0000 s	0.2798 s
Server S1 in DMZ	0.0000 s	0.0633 s	0.0000 s
Server S2 in DMZ	0.0000 s	0.1898 s	0.0000 s
DMZ production servers	0.0000 s	0.0000 s	0.1492 s
General office assets	0.0000 s	0.0000 s	0.0373 s
Total time (full session)	0.28 s	0.6329 s	0.9325 s

those that have been estimated, we evaluated the maximum number of clients (sessions) with different authorization permissions the server is able to handle within an hour having 90% of CPU load. Our assessment is quite straightforward; knowing that, for the existing server, it takes 0.28 seconds to handle user assigned *role 1* and using the simplified formula for CPU utilization, one can easily calculate the number of served clients (considering given conditions and using (7)):

$$\text{users} = \frac{90 \cdot 3\,600}{100 \cdot 0.28}, \tag{8}$$

which results in about 11 571 handled employees per hour (Figure 3). Assuming the linear growth, one can estimate that, during two hours, the machine is able to manage about $11\,571 \cdot 2 = 23\,142$ connections, while through three hours this value increases to $11\,571 \cdot 3 = 34\,713$ handled users (and so on). The same type of assessment was performed for the remaining roles (*role 2* and *role 3*).

The performed time analysis confirmed that the server is capable of dealing with the maximum of 3 474 users within an hour having 90% of CPU load (handling employees assigned *role 3* permissions). Assuming the linear growth of the number of served users, one can say that within 24 hours single server is able to manage $24 \cdot 3\,474 = 83\,376$ employees having *role 3* privileges. When the company decides to increase the number of handled *most secure* users' connections by, for instance, two times, and keep the CPU load at the same level simultaneously, there is a need for another server to handle them all. Managing $83\,376 \cdot 2 = 166\,752$ employees assigned *role 3* permissions greatly exceeds server capabilities. To serve 166 752 clients, single server would have to work for two days, not as we want it to be, for 24 hours. However, when we change the protection type to TLSv1 leaving the CPU load on the same level, it then becomes possible to handle all connections without buying new equipment, since the maximum number of users having *role 1* privileges (277 704), who can be dealt with in 24 hours, is greater than the number of employees assigned *role 3* permissions served by server during two days. Using simple math we can say that during

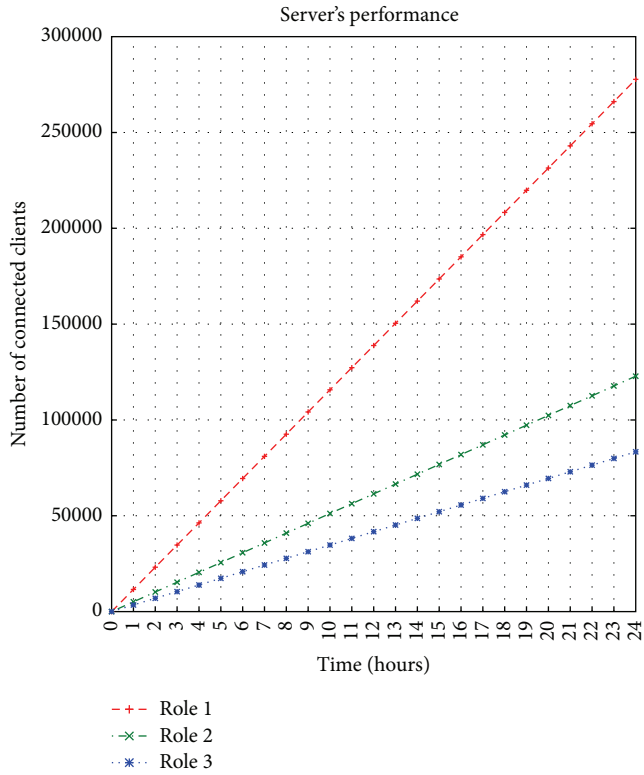


Figure 3: Server's performance in the considered *scenario*.

24 hours server which deals with *role 1* permissions set is able to manage roughly 67% more connections than the one which works for 48 hours with users having *role 3* privileges (or, in other words, server handling users of the third role is capable of realizing about 40% less connections).

Analysing obtained results and assuming 90% of machine load, one can easily notice that the server is able to handle clients with the first authorization level faster than the same number of users permitted to perform *role's 3* actions. Gathered results clearly indicate the relationship between the assigned role and consumption of server resources: the longer time the action needs to accomplish, the more the server resources are going to be used. The server, which works longer, utilizes more resources, thereby consuming a greater amount of energy.

The Energy Usage Analysis. Besides the presented time analysis, we modeled the energy usage for our scenario. We collected the data required by the AQoPA's Energy Analysis Module from the official documentation of the CPU which was used for the metrics generation [33]. In our analysis, we considered only the *busy* time of the CPU. We simply ignored the left-over time and focused only on power consumed while performing security operations. Table 7 contains the results, expressed in joules, gathered by the AQoPA tool.

The QoP Analysis. In [31] different versions of the TLS protocol were analysed: their security *quality* was assessed in terms of *confidentiality*, *integrity*, *availability*, and *authorization*.

Table 7: Server's energy analysis results obtained by the AQoPA's energy analysis module (joules).

Action performed (access)	Scenario RBAC role		
	Role 1	*Role 2*	*Role 3*
E-mail	2.1284 J	6.0129 J	8.8610 J
FTP	2.6605 J	6.0129 J	4.4305 J
Web (WWW)	0.7981 J	2.4051 J	3.5444 J
Application(s)	1.0642 J	3.6077 J	5.3166 J
Data center	6.6512 J	0.0000 J	13.2915 J
Server S1 in DMZ	0.0000 J	3.0064 J	0.0000 J
Server S2 in DMZ	0.0000 J	9.0194 J	0.0000 J
DMZ production servers	0.0000 J	0.0000 J	7.0888 J
General office assets	0.0000 J	0.0000 J	1.7722 J
Total power usage (full session)	13.3024 J	30.0644 J	44.305 J

Table 8: The qualitative interpretation of QoP evaluation of the analysed versions of TLS protocol.

Security mechanisms	Security attribute		
	Confidentiality	Integrity	Availability
RC4 + MD5	Very low	Very low	Very high
AES/CBC 128 + SHA-1	High	Medium	Medium
AES/CBC 256 + SHA-512	Very high	Very high	Very low

Authors presented the quality of protection evaluation of the chosen security mechanisms and defined logical, consistent steps that help assess the quality of security operations. Since in their work researchers evaluated two of our four TLS versions, we summarized their results in Table 8, adding the qualitative interpretation of QoP evaluation, computed for the remaining cases with the help of the SME tool. The complete evaluation of the chosen TLS versions was presented in [31]. The following examination is based on the analysis performed in [31].

When the analysis is finished, one can finally interpret the results. As is apparent from Table 8, TLS version 1 (RC4 + MD5) stands for the most CPU-efficient solution. The crucial part of the TLS protocol, expressed by the means of the *authorization* attribute, indicates that AES/CBC 256 + SHA-512 and AES/CBC 256 + SHA-1 are the best options when it comes to the server authorization. Comparing the results obtained with the SME tool, one can choose those security mechanisms, which satisfy given requirements best in terms of *confidentiality*, *integrity*, and *availability*.

New Type of Analysis: The Financial Analysis. In this section, we present a brief overview of predicting the total budget required to manage a data center, mainly focusing on a method for measuring its total cost and indicating possible gains. In addition, we try to confirm our thesis about the influence of the security management and data flow on the total cost of the data center maintenance by the analysis of the defined scenario, together with the estimation of its accompanying costs. Performing our estimations, we utilized formulas presented in Section 5.

Determining the total cost of the working machine, the biggest attention should be drawn to the CPU, since this component is the largest energy consumer in a typical computer. Using (1), one is able to estimate the daily, weekly, monthly, and annual cost of power consumption for each CPU in the enterprise. Although the CPU is clearly the largest consumer of energy, total energy utilization of a single server consists of power usage of all its components. To obtain total power consumption of a single server, one needs to take into consideration the power usage of its HDDs, RAMs, network cards, video cards, and so forth. Depending on the server's workload, the energy utilized for the cooling system (the amount of the power needed to ensure a proper temperature for the busy machine and its components) must be covered as well. Let us have a closer look at, for instance, CPU's fan power consumption. One should keep in mind that *idle* server requires less energy and thus cooling than the *busy* one. In some certain scenarios CPU fans contribute much to the power requirements of PC components. Among factors that affect fan energy usage one can enumerate the fan speed (measured in RPM), fan size (80 mm, 92 mm, 120 mm, 140 mm, and 200 mm), and the fact whether it has LED lights. Typical power consumption of a 80 mm case fan (3,000 RPM) ranks between 2.4 and 3 W [34].

As it can be seen from the time analysis, to handle the exact number of incoming connections, the server needs to spend more time on serving users assigned *role 3* privileges, than those having *role 1* permissions. Based on Figure 3, to deal with about 20 000 users, the machine works through approximately 1.7 hours (*role 1*), but, considering *role 3* accesses, this value increases to about 5.8 hours. Suppose that our single server uses CPU together with 3 W fan (both working for twenty-four hours a year). During 5.8 *busy* hours, fan utilizes about $(5.8 \cdot 3\,\text{W})/1\,000 = 0.0174$ kilowatt-hours (which costs approximately 0.00105966$), while processing *role 1* requests for 1.7 hours requires about three times less money (approximately $(1.7 \cdot 3\,\text{W})/1000 = 0.0051$ kWhs, 0.00031059 dollars).

The conclusion is that the role-switching (the security level switching) has undeniable impact not only on CPU energy utilization but also on the amount of power consumed by cooling and, in fact, on all the working components of the machine, which strictly translates into cost savings. In this section, however, we focused only on the energy usage of the CPU as the main unit utilized for the cryptographic operations and made only a brief note about its influence on remaining energy consumers. Power consumption of other server components can be found on the manufacturer's websites and on the websites which perform independent hardware benchmarks and tests. The analysis of the optimal energy consumption of a single server along with all its components is left for the future work. Nevertheless, performed research provides serious argumentation to believe that the reduction of the CPU usage, and thus the amount of the utilized energy, entails significant economic profits. As stated before, cost savings are in fact even higher than those estimated here, since, saving the CPU power usage, we reduce the amount of energy utilized for the cooling system, as well as decreasing power consumption of all

the server's physical components. Our study showed that, ensuring required security, it is possible to reduce the power consumption and increase cost savings at the same time. At first glance, figures presented here may seem irrelevant; however, when, put in the context of a large data center environment, they can quickly become very significant.

As it might seem on the basis of the so far performed analyses, the total cost of the data center utilization does not depend only on the power consumption of the server's CPU. Calculating the total cost of operating a data center one needs to take into account both fixed and variable costs, which are affected by complex and interrelated factors. From the economic analysis of the power consumed by the central processing unit, one can note that, with the exact number of served users, it is possible to save money, only by switching the level of protection from the *strongest* to the *weakest* one. The financial analysis performed earlier proved that, reducing protection mechanisms, one can expect significant financial profits. Analysing results estimated during our research, one can see that changing the level of protection it is possible to handle the required number of users and increase financial profits even more (since the reduction of the CPU load implicates the decrease of the power used by all server components, resulting in lower costs). However, in the economic stage of the analysis process, we need to look at power consumption, heating and cooling, and the data center footprint.

Determining the approximate, total cost of a whole data center, we assumed to use Dell PowerEdge R710 servers and supposed that the average lifetime of a single machine is equal to about three years. However, we did include neither the network nor the storage footprint (nor its equipment) in our estimation. Server-specific numbers (such as BTU) utilized in our evaluation were obtained from the technical guide of the Dell PowerEdge R710 server.

Although it does not seem to result directly from the economic analysis of a single machine, the presented solution can bring real, meaningful cost savings. In our approach economic profits actually come from the number of working machines together with its load factor. Modification in the configuration of utilized security mechanisms lets one obtain significant benefits. Presented approach brings additional possibilities: by switching security mechanisms from the *strongest* to the *weakest*, it is possible to provide effective services and maximize the utilization of hardware resources at the same time. Since we can accomplish exact goal using *weaker* security mechanisms, in many situations it is wasteful to assign too many hardware resources to perform the given task. Applying the proposed solution to the existing IT environment, one can observe serious reduction in IT costs while increasing the efficiency, utilization, and flexibility of their existing computer hardware. Table 9 explores this concept in more detail.

As it was proved by the time analysis, server working with *role 3* permissions is able to handle about 3 474 users within an hour having 90% of CPU load. Since we assumed that the number of users grows linearly, within 24 hours, it gives us $3\,474 \cdot 24 = 83\,376$ employees a day, resulting in $83\,376 \cdot 365 = 30\,432\,240$ connections a year per server. If we assume that we have at our disposal the whole data center, it will turn out that we can serve roughly $30\,432\,240 \cdot 520 = 15\,824\,764\,800$ users assigned *role 3* permissions a year. To handle exact number of users assigned *role 1* privileges with 90% of CPU load and being aware that using *role 1* permissions server is capable of dealing with 11 571 users within an hour, it is trivial to compute number of physical machines capable of managing quite same number of employees, with different security mechanisms applied:

Table 9: CPU load equals 90%; number of connections to handle is given for the scenario. Estimated values represent the total annual cost of the energy and cooling usage, rounded up to the nearest dollar.

Scenario (users to handle ≈ 15 824 764 800)		
	Server(s)	$\varsigma_{\text{power+cooling}}$
Role 1	156	57 051$
Role 2	352	128 728$
Role 3	520	190 168$

$$\text{users}_{\max_{S1R1}} \cdot \chi \cdot \rho \cdot \mu_{S1R1} \approx \text{users}_{\max_{S1R3}} \cdot \chi \cdot \rho \cdot \mu_{S1R3}, \quad (9)$$

where $\text{users}_{\max_{S1R1}}$ is the maximum number of users assigned *role 1* permissions that can be handled within an hour by the single server, $\text{users}_{\max_{S1R2}}$ is the maximum number of users assigned *role 3* permissions that can be handled within an hour by the single server, χ is the total amount of hours when the single server was busy, ρ is the total amount of days when the single server was busy, μ_{S1R1} is the number of servers being capable of dealing with the given traffic (supposing managing $\text{users}_{\max_{S1R1}}$ employees an hour, having *role 1* privileges through χ hours for ρ days), and μ_{S1R3} is the number of servers being capable of dealing with the given traffic (supposing managing $\text{users}_{\max_{S1R3}}$ employees an hour, having *role 3* privileges through χ hours for ρ days), which, in our case, results in

$$11\,571 \cdot 24 \cdot 365 \cdot \mu_{S1R1} \approx 3\,473 \cdot 24 \cdot 365 \cdot 520, \quad \mu_{S1R1} \approx 156. \quad (10)$$

From Table 9, it can be seen that, to handle about 15 824 764 800 connections (a year) with *role 3* permissions assigned, it is necessary to utilize all physical machines in the enterprise. However, exact amount of employees can be dealt with by about 156 servers, if we change the communication channel protection type to *role 1* privileges set. Thus, it is clear that power delivery and cooling savings can increase to roughly 70%.

As assumed earlier in the paper, performing day-to-day tasks, single server consumes on average about 0.3 kilowatts per hour. According to our model and analysis performed by the AQoPA tool, while processing defined security operations, CPU used up roughly about 0.04 kilowatts for each role, per all sessions within an hour, having 90% of CPU load. Using (1) and assuming that the cost of cooling is approximately equal to the cost of consumed energy, it is quite straightforward to estimate the total, annual cost of

Table 10: Approximate annual cost of energy consumption (in US dollars). Number of handled users is equal to 33 288 000 per server a year; CPU load varies between the roles.

Server(s)	Scenario: RBAC role *Role 1* (CPU load ≈ 29.55%)	*Role 2* (CPU load ≈ 66.80%)	*Role 3* (CPU load ≈ 98.43%)
	$\varsigma_{\text{power annual}} + \varsigma_{\text{cooling annual}}$	$\varsigma_{\text{power annual}} + \varsigma_{\text{cooling annual}}$	$\varsigma_{\text{power annual}} + \varsigma_{\text{cooling annual}}$
1	335	354	370
13	4 356	4 601	4 810
520	174 238	184 054	192 394

Table 11: Approximate annual environmental impact (in pounds of CO_2). Number of handled users is equal to 33 288 000 per server a year; CPU load varies between the roles.

Server(s)	Scenario: RBAC role *Role 1* (CPU load ≈ 29.56%)	*Role 2* (CPU load ≈ 59.36%)	*Role 3* (CPU load ≈ 98.45%)
1	5 117	5 405	5 650
13	66 519	70 267	73 451
520	2 660 770	2 810 679	2 938 038

the physical machine dealing with defined scenario. Table 10 comprises assessed costs for single machine, a rack, and the whole data center.

Being aware of the annual usage cost of the single server, we were hereby able to compute the amount of money spent on any number of working machines.

In conclusion, such security mechanisms switching can offer a variety of economic advantages: it permits one to increase the scale of server infrastructure without purchasing additional pieces of hardware and allows resources to be used more efficiently. In addition to savings in hardware costs, security level switching decreases the amount of floor space and maintenance expenditures. Such server consolidation also reduces the overall footprint of the entire data center. That means far fewer servers, less networking gear, a smaller number of racks needed, all of which translates into less data center floor space required. Consolidating server onto far fewer physical machines means lowering monthly power and cooling costs in the data center.

New Type of Analysis: The Carbon Dioxide Emissions Analysis. The energy usage estimation performed for the example call center can be a good start for the research on the efficient carbon dioxide emissions. The following analysis confirms the statement that the more energy we save, the less CO_2 our machine will produce. When it comes to the emissions of CO_2 analysis of an example IT environment, we proposed using (6) to estimate its produced volume. Values estimated with the help of (6) are summarized in Table 11. They refer to the single physical machine (the rack and the data center), performing actions defined in our security-based data flow management RBAC model, and represent the total annual environmental impact of server's energy usage, rounded up to the nearest pound.

Table 12: CPU load equals 90%; number of connections to handle is given for the defined scenario. Estimated values represent the total, approximate annual environmental impact (in pounds of CO_2), rounded up to the nearest pound.

Scenario (users to handle ≈ 15 824 764 800)		
	Server(s)	Pounds of CO_2
Role 1	156	871 218
Role 2	352	1 965 790
Role 3	520	2 904 045

Same as in the case of the financial and economic analyses, it is possible to estimate the amount of the carbon dioxide, which can be saved if only we perform the proposed security-switching. As it is apparent from Table 12, changing protection mechanisms from the *strongest* to the *weakest* ensuring required security and fulfilling all the security objectives at the same time, it is possible to decrease the amount of the carbon dioxide released to the atmosphere by about 9% (roughly pounds) a year per data center.

According to [35], the average amount of released carbon dioxide to the atmosphere per kWh was about 0.84 368 kg (1.86 pounds) in April 2014; therefore a single working server equipped with the considered CPU produced roughly 5 585 pounds of carbon dioxide a year (having ≈90% of CPU load and handling 3474 users assigned third role in the considered scenario). Regarding the whole data center, it released about 2 904 045 pounds of carbon dioxide to the atmosphere, which can be simply translated to 1 317 253 kilograms of CO_2. Comparing the above estimated values for *role 3* to those assessed when dealing with the first one, it can be seen that switching between the *strongest* and the *weakest* security mechanisms one can reduce the emissions of CO_2 by ≈9%,

that is, about 277 268 pounds (125 767 kilograms) per data center.

In accordance with the information at [36], the annual amount of released carbon dioxide saved by switching between the roles (from *role 3* to *role 1*) for the data center is equivalent to about 27 passenger vehicles, 14 152 gallons of gasoline consumed, 135 088 pounds of coal burned per year, or 17.3 homes' electricity use for one year. However, bear in mind that the data presented here correspond to the average values. Let us have a closer look at, for instance, number of gallons of gasoline consumed (e.g., by the passenger vehicles). Here, the number of gallons of gasoline consumed depends on many different factors, that is, the type of the gearbox (automatic or manual), engine size, or even the weight and the shape of the car. More profound analysis requires consideration of all of these factors.

Steps 4 and 5 (recommendation and implementation). To serve as a specialized solution for secure systems, our decision support system took as an input characteristics of the environment's traffic, that is, the number of established connections, available services, utilized protocols along with their configuration details, and all the possible security mechanisms that can be used in the considered environment (encryption, decryption, hashing algorithms, and many more). Besides the mentioned components, to obtain the proper configuration of available parameters that is the most effective in given circumstances, one needs to provide hardware metrics as well. Since usually complex IT environments are characterized by high dynamism, it is reasonable to define the approximate time of the analysis (a month, a year, with, for instance, one-hour interval, when the traffic characteristic can change and the whole analysis process can be repeated). The goal of the system is the efficient management of the incoming connections in terms of utilized physical resources, system load, financial costs, CO_2 emissions, and the amount of consumed power. At the output, one receives the estimated number of physical machines capable of handling given traffic together with their assessed load. Results of the remaining analyses are available as well. Retrieved information enables users to find the most appropriate solution to their problems. Let us summarize advices suggested by our DSS for the proposed case study. When it comes to the time analysis, the conclusion is simple: if one is interested in security mechanisms that are the most *time-effective*, our decision support system indicates that TLSv1 is the best option. Looking for the security solution with the *lowest energy consumption*, our DSS suggests that RC4 and MD5 are the right choice. The variety of alternatives examined by the QoP analysis facilitates decision-making process as well. For instance, if we do care about the *confidentiality*, *integrity*, and *availability*, TLSv3 achieves the strongest possible security level (it is in fact *the most secure* among others). From the economic point of view, TLSv1 seems to be a quite interesting alternative: designed decision support system suggests, that if we choose RC4 and MD5 instead of AES/CBC 128 and SHA-1 or AES/CBC 256 and SHA-512 as utilized security mechanisms, we will save about 18 156 dollars per data center on average a year.

6.2. Security Objectives Realization. By choosing the role-based-access control model as the example realization of the data flow management, we answered questions about the *security management* and *access control* (SO1). We defined three roles, which differ in applied security mechanisms, and mapped them to established enterprise functions, in order to *provide management direction and support for information security in accordance with business requirements and relevant laws and regulations* [29].

As mentioned before, the TLS communication channel protection allowed us to fulfill the *confidentiality, authenticity, or integrity protection* security objective (SO2). While securing the communication channel, we utilized cryptographic algorithms and usage practices selected according to best practice.

Regarding the *system capacity management* (SO3), we performed the multilevel analysis according to the proposed DSS. We estimated and compared number of users that the server can handle during an hour, managing connections protected by different versions of the TLS protocol, and determined time and energy consumed by established sessions, along with assessing their economic and environmental impact. Such an approach allows us to adjust the security level to the capacity of the system while meeting defined security objectives. Performed, multidimensional analysis allows us to properly examine system capabilities and select those security mechanisms, which ensure security objectives realization together with decreasing financial costs and reducing negative environmental impact.

When it comes to the *assurance of correct and secure operation of information processing and handling facilities* (SO4), we modeled subprocesses responsible for secure information management: `AccessFTPServer`, `AccessWebServer`, `AccessAppServer`, `AccessServerS1`, `AccessServerS2`, `AccessGeneralOfficeAssets`, and `AccessDMZProductionServers`.

To maintain *the integrity and the availability of information* (SO5), we proposed modeling data center access (subprocess `AccessDataCenter`), which can simply be mapped to back-up process in the proposed, example implementation environment. The `AccessDataCenter` subprocess uses channel encryption to ensure the security of the operation and stores the back-up data at a secure, remote location.

Meeting the sixth security objective (SO6), we modeled an `AccessEmailServer` subprocess, which is responsible for providing a secure electronic message exchange.

Based on the gathered results, it is possible to implement the solution which satisfies given requirements best. It is also worth mentioning that each of the proposed roles (regardless of utilized security mechanisms) fulfills all the defined security objectives.

7. Conclusions

In the paper we used Quality of Protection Modeling Language to prepare the model of example business scenario for the enterprise having role-based access control management implemented. We defined a scenario with dissimilar levels

of security and investigated the performance of the server handling miscellaneous number of users with different RBAC roles assigned. On the basis of the gathered results, we indicated that the user access control management has a meaningful impact on overall system's performance. Our research proved that drawing attention to the system's efficiency while implementing the role-based access control policy is crucial from the user access control point of view, usability, and security management. Furthermore, the ability of preparing the access control management security model in Quality of Protection Modeling Language confirmed its extensibility and flexibility with the role-based access control functionality. Our analysis showed that it is possible to achieve a balance between security and performance.

In addition, on the basis of the performed analyses, we proposed the foundations of the model-driven decision support system that helps one make the decision and determine which of the available security mechanisms meets given requirements best. Using introduced system, one can choose between several decision-making techniques, which can simply be translated into different methods of analysis. With the presented framework it is possible to find the fastest, the cheapest, the most secure, the most energy-efficient, or the most environment friendly solution and obtain estimated results that one can actually compare.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work is supported by the Polish National Science Centre Grant 2012/05/B/ST6/03364.

References

[1] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 68–73, 2009.

[2] J. G. Koomey, *Estimating total power consumption by servers in the U.S. and the world [Ph.D. thesis]*, 2007.

[3] C. D. Patel and A. J. Shah, *Cost Model for Planning, Development and Operation of a Data Center*, HP Laboratories, Palo Alto, Calif, USA, 2005.

[4] The official web page of the QoPML project, 2012, http://www.qopml.org.

[5] A. O'Connor and R. Loomis, "Economic analysis of role-based access control," Tech. Rep., National Institute of Standards and Technology, 2010.

[6] "A comparison of security analysis techniques for RBAC models," in *Proceedings of the 2nd Annual CCWIC*, 2010.

[7] A. K. Agarwal and W. Wang, "On the impact of quality of protection in wireless local area networks with IP mobility," *Mobile Networks and Applications*, vol. 12, no. 1, pp. 93–110, 2007.

[8] B. Ksiezopolski and Z. Kotulski, "Adaptable security mechanism for dynamic environments," *Computers & Security*, vol. 26, no. 3, pp. 246–255, 2007.

[9] E. LeMay, W. Unkenholz, D. Parks, C. Muehrcke, K. Keefe, and W. H. Sanders, "Adversary-driven state-based system security evaluation," in *Proceedings of the 6th International Workshop on Security Measurements and Metrics (MetriSec' 10)*, ACM, September 2010.

[10] S. Lindskog, *Modeling and tuning security from a quality of service perspective [Ph.D. thesis]*, Chalmers University of Technology, Gothenburg, Sweden, 2005.

[11] A. Luo, C. Lin, K. Wang, L. Lei, and C. Liu, "Quality of protection analysis and performance modeling in IP multimedia subsystem," *Computer Communications*, vol. 32, no. 11, pp. 1336–1345, 2009.

[12] C. S. Ong, K. Nahrstedt, and W. Yuan, "Quality of protection for mobile multimedia applications," in *Proceedings of the International Conference on Multimedia and Expo (ICME '03)*, vol. 2, pp. 137–140, Baltimore, Md, USA, July 2003.

[13] D. C. Petriu, C. M. Woodside, D. B. Petriu et al., "Performance analysis of security aspects in UML models," in *Proceedings of the 6th International Workshop on Software and Performance (WOPS '07)*, pp. 91–102, ACM, February 2007.

[14] P. A. Schneck and K. Schwan, "Authenticast: an adaptive protocol for high-performance, secure network applications," Tech. Rep., Georgia Institute of Technology, 1997.

[15] Y. Sun and A. Kumar, "Quality-of-protection (QoP): a quantitative methodology to grade security services," in *Proceedings of the 28th International Conference on Distributed Computing Systems Workshops (ICDCS '08)*, pp. 394–399, IEEE Computer Society, Beijing, China, June 2008.

[16] U. R. F. Averweg, *Decision-Making Support Systems: Theory & Practice*, bookboon.com, 2012.

[17] D. J. Power, "A brief history of decision support systems," http://dssresources.com/history/dsshistory.html.

[18] D. J. Power and R. Sharda, "Model-driven decision support systems: concepts and research directions," *Decision Support Systems*, vol. 43, no. 3, pp. 1044–1061, 2007.

[19] Z. Shi, "Knowledge-based decision support system," *Journal of Computer Science and Technology*, vol. 2, no. 1, pp. 22–29, 1987.

[20] R. Matulevičius, H. Lakk, and M. Lepmets, "An approach to assess and compare quality of security models," *Computer Science and Information Systems*, vol. 8, no. 2, pp. 447–476, 2011.

[21] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Computer role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, 1996.

[22] T. Lodderstedt, D. Basin, and J. Doser, "SecureUML: a UML-based modeling language for model-driven security," in *(UML) 2002—The Unified Modeling Language*, vol. 2460 of *Lecture Notes in Computer Science*, pp. 426–441, Springer, Berlin, Germany, 2002.

[23] J. Jürjens, *Secure System Development with UML*, Springer, New York, NY, USA, 2007.

[24] N. Sklavos, P. Kitsos, K. Papadopoulos, and O. Koufopavlou, "Design, architecture and performance evaluation of the wireless transport layer security," *The Journal of Supercomputing*, vol. 36, no. 1, pp. 33–50, 2006.

[25] A. Stubblefield, A. D. Rubin, and D. S. Wallach, "Managing the performance impact of web security," *Electronic Commerce Research*, vol. 5, no. 1, pp. 99–116, 2005.

[26] J. Jürjens, "Security and compliance in clouds," in *Proceedings of the 4th Pan-European Conference*, IT-Compliance, 2011.

[27] R. M. Savola, "Quality of security metrics and measurements," *Computers & Security*, vol. 37, pp. 78–90, 2013.

[28] B. Ksiezopolski, "QoP-ML: quality of protection modelling language for cryptographic protocols," *Computers & Security*, vol. 31, no. 4, pp. 569–596, 2012.

[29] International Standard ISO/IEC 27002, "Information technology—security techniques—code of practice for information security controls," 2013.

[30] K. Mazur, B. Ksiezopolski, and Z. Kotulski, "The robust measurement method for security metrics generation," *The Computer Journal*, 2014.

[31] B. Ksiezopolski, T. Zurek, and M. Mokkas, "Quality of protection evaluation of security mechanisms," *The Scientific World Journal*, vol. 2014, Article ID 725279, 18 pages, 2014.

[32] D. Rusinek, B. Ksiezopolski, and A. Wierzbicki, "Security trade-off and energy efficiency analysis in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 943475, 17 pages, 2015.

[33] http://ark.intel.com/products/52577/Intel-Xeon-Processor-X5675-12M-Cache-3_06-GHz-6_40-GTs-Intel-QPI.

[34] R. Chheda, D. Shookowsky, S. Stefanovich, and J. Toscano, "Profiling energy usage for efficient consumption," *The Architecture Journal*, vol. 18, pp. 24–27, 2008.

[35] U.S. Energy Information Administration, "How much carbon dioxide is produced per kilowatthour when generating electricity with fossil fuels?" http://www.eia.gov/tools/faqs/faq.cfm?id=74&t=11.

[36] Inventory of U.S. Greenhouse Gas Emissions and Sinks: 1990–2011, Chapter 3 (Energy), Tables 3-12, 3-13, and 3-14, U.S. Environmental Protection Agency, Washington, DC, USA, Inventory of U.S. Greenhouse Gas Emissions and Sinks: 1990–2011, Annex 2, Methodology for estimating CO2 emissions from fossil fuel combustion, Table A-36. U.S., Environmental Protection Agency, Washington, DC, USA, greenhouse gas equivalencies calculator, http://www.epa.gov/cleanenergy/energy-resources/calculator.html.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 823609, 15 pages
http://dx.doi.org/10.1155/2015/823609

Research Article

Multicriteria Resource Brokering in Cloud Computing for Streaming Service

Chih-Lun Chou,[1] Gwo-Jiun Horng,[2] Chieh-Ling Huang,[3] and Wei-Chun Hsieh[4]

[1]*Department of Information Telecommunications Engineering, Ming Chuan University, Taoyuan 33348, Taiwan*
[2]*Department of Computer Science and Information Engineering, Southern Taiwan University of Science and Technology, Tainan 71005, Taiwan*
[3]*Department of Computer Science and Information Engineering, Chang Jung Christian University, Tainan 71101, Taiwan*
[4]*Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 70101, Taiwan*

Correspondence should be addressed to Chieh-Ling Huang; kaio.hcl@gmail.com

Received 27 October 2014; Accepted 26 March 2015

Academic Editor: Jian Guo Zhou

By leveraging cloud computing such as Infrastructure as a Service (IaaS), the outsourcing of computing resources used to support operations, including servers, storage, and networking components, is quite beneficial for various providers of Internet application. With this increasing trend, resource allocation that both assures QoS via Service Level Agreement (SLA) and avoids overprovisioning in order to reduce cost becomes a crucial priority and challenge in the design and operation of complex service-based platforms such as streaming service. On the other hand, providers of IaaS also concern their profit performance and energy consumption while offering these virtualized resources. In this paper, considering both service-oriented and infrastructure-oriented criteria, we regard this resource allocation problem as Multicriteria Decision Making problem and propose an effective trade-off approach based on goal programming model. To validate its effectiveness, a cloud architecture for streaming application is addressed and extensive analysis is performed for related criteria. The results of numerical simulations show that the proposed approach strikes a balance between these conflicting criteria commendably and achieves high cost efficiency.

1. Introduction

As an emerging technology, cloud computing combines various computing paradigms and provides main service models which are known as SaaS (Software as a Service), PaaS (Platform as a Service), and IaaS (Infrastructure as a Service). Among these three service models, IaaS arouses the outsourcing trend of IT infrastructure by provisioning fundamental computing resources where the consumers are able to deploy and run the desired applications as specific services. Most of the IaaS consumers are providers of Internet application hosting complex service-based platforms. Those service platforms require efficient resource allocation to ensure their particular QoS (Quality of Service) which is usually specified in SLA (Service Level Agreement) and meanwhile prevents overprovisioning in order to reduce operating costs. On the other hand, providers of IaaS also concern their profit performance and energy consumption while providing these virtualized resources. Therefore in this paper, a cloud architecture for streaming service platform is addressed and an efficient resource brokering approach based on the analysis of both objectives is proposed.

Inheriting the essence of distributed and parallel characteristics in grid computing and with the growth of virtualization as well as developed web services technologies, cloud computing becomes the most promising computing and service paradigm. The National Institute of Standards and Technology (NIST) has released the definition of it [1]. Cloud computing is a model for enabling ubiquitous, convenient, and on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. The cloud model defined in [1] also outlines five essential characteristics and three service models. The three service models are SaaS, PaaS,

Table 1: Amazon EC2 pricing table.

	Linux/UNIX usage	Windows usage
Standard on-demand instances		
Small (default)	\$0.085 per hour	\$0.12 per hour
Large	\$0.34 per hour	\$0.48 per hour
Extra large	\$0.68 per hour	\$0.96 per hour
Micro on-demand instances		
Micro	\$0.02 per hour	\$0.03 per hour
Hi-memory on-demand instances		
Extra large	\$0.50 per hour	\$0.62 per hour
Double extra large	\$1.00 per hour	\$1.24 per hour
Quadruple extra large	\$2.00 per hour	\$2.48 per hour
Hi-CPU on-demand instances		
Medium	\$0.17 per hour	\$0.29 per hour
Extra large	\$0.68 per hour	\$1.16 per hour

and IaaS. SaaS lets consumers use the provider's applications running on a cloud infrastructure, for example, Gmail and Google Docs. Consumers generally access those applications via a web-based interface such as a browser from various thin client devices. PaaS and IaaS are similar in a manner. They both provide consumers computing resources like hardware (e.g., servers, networks, and storage) and software (operating systems, and databases). Consumers of PaaS can use programming languages and tools supported by the PaaS providers to create specific applications and deploy them onto the particular cloud platform of PaaS providers, while the consumers of IaaS providers are able not only to deploy and run arbitrary applications but also to setup their desired environment such as operating systems and runtime libraries.

Indubitably, the most significant core in cloud computing technologies is virtualization [2]. By utilizing virtualization in a datacenter, processing, storage, networks, and other computing resources are provided to cloud consumers with ease. This resource pooling characteristic derives the service model so-called Infrastructure as a Service (IaaS). Providers of IaaS such as Amazon EC2 [3] pack fundamental computing resource in the form of virtual machines (VMs) and let the cloud consumers deploy and run arbitrary software, which can include operating systems and applications. Due to the on-demand and rapid elasticity, the outsourcing of computing resources with providers of IaaS is quite beneficial for cloud consumers like various providers of Internet application.

However, this outsourcing trend arises a key issue of resource allocation owing to pay-per-use business model of IaaS like Table 1. Consumers are charged for their usage of these IaaS services.

Consumers pay for compute capacity by the hour with no long-term commitments. This frees consumers from the costs and complexities of planning, purchasing, and maintaining hardware and transforms what are commonly large fixed costs into much smaller variable costs. The pricing in Table 1 includes the cost to run private and public AMIs on the specified operating system. Pricing is per instance-hour consumed for each instance, from the time an instance is launched until it is terminated. Each partial instance-hour consumed will be billed as a full hour.

The concept is from Utility Computing and Services Computing. In fact, after water, electricity, gas, and telephony, there is an increasingly perceived vision that computing will be the 5th utility one day. Thus, a resource provisioning mechanism that both prevents underprovisioning in order to assure QoS via Service Level Agreement (SLA) and avoids overprovisioning so as to reduce cost becomes a crucial priority and challenge in the design and operation of complex service-based platforms such as streaming service. On the other hand, cloud infrastructure providers also concern their profit performance and energy consumption. Profit performance is the profitability. Sustaining profitability and growth is the main target for these companies and corporations. Energy consumption is a critical topic not only in environmental issues but also in reducing operating costs for electricity. Those criteria for both service-based platforms and cloud infrastructure providers are important and may be conflicting.

In this paper, contemplating both service-oriented and infrastructure-oriented criteria, we regard this resource allocation problem as Multicriteria Decision Making problem. Our intention is to design a trade-off based strategy and propose an effective resource provisioning algorithm for an autonomous resource broker in the cloud, as shown in Figure 1. After the streaming service platform evaluates the requirements and defines a mapping between the requests' service level requirements and resource level requirements, the autonomous resource broker will regulate the supply and demand of cloud resources between the streaming service platform and the cloud infrastructure provider based on not only incoming requests of the streaming service but also both the service-oriented criteria which are in the form of SLA and infrastructure-oriented criteria. The solution of the algorithm is obtained by formulating and solving a goal programming model. In particular, a cloud architecture for streaming application is addressed as well as extensive analysis and experiments are performed for related criteria. The results of numerical simulations show that the proposed approach strikes a balance between these conflicting criteria commendably and achieves high cost efficiency.

The rest of this paper is organized as follows. In Section 2, related works are reviewed. The system model and assumption of cloud architecture for streaming service are described in Section 3. Section 4 details the analysis of service-oriented and infrastructure-oriented criteria. In Section 5, the goal programming model and algorithm are depicted. Simulations are presented in Section 6. Finally, Section 7 summarizes our conclusions and outlines our future works.

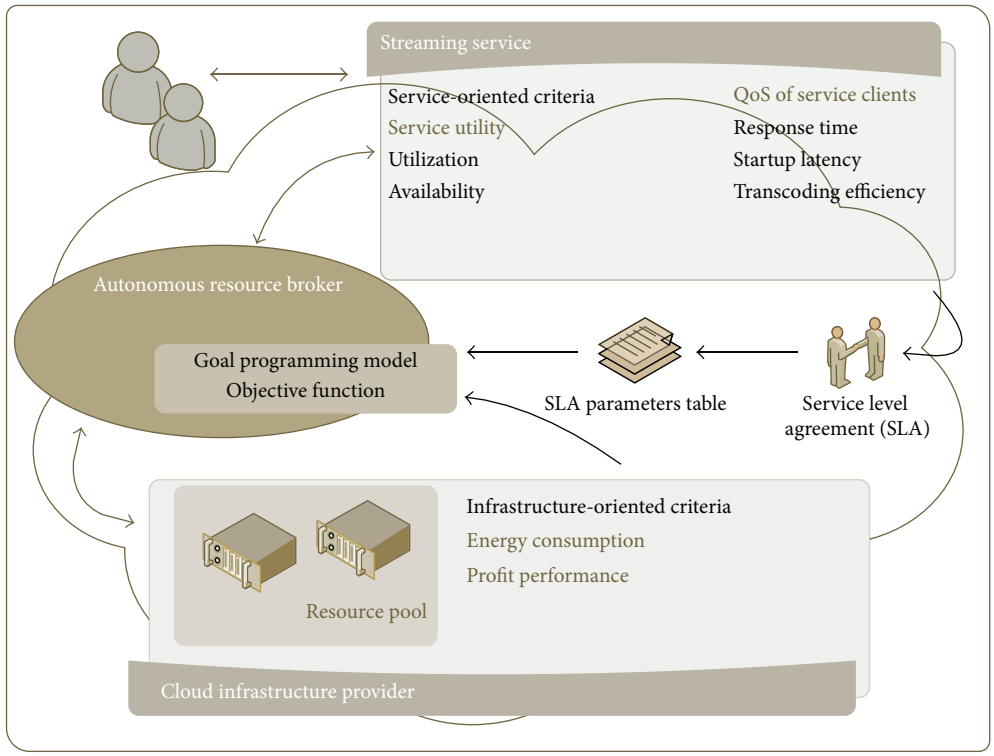


Figure 1: Overview of resource brokering for streaming service in the cloud.

2. Related Works

There are many research papers probing into resource allocation and provisioning in cloud computing. Here we can sum up the three main steps of resource allocation in cloud.

2.1. Evaluate the Requirements. Evaluate and define a mapping between service level requirements and resource level requirements. Service level requirements are generally defined in SLA based on specific parameters such as availability, response time. Resource level requirements are often outlined as cores, memory, bandwidth, and so forth. This step also requires performance and capacity modeling.

2.2. Resource Brokering and Provisioning. This is the main step of the whole resource allocation in cloud since the determination of the number of cloud resources such as virtual machines reflects not only the efficiency about the utilization, QoS, and operating cost of cloud infrastructure consumers but also the performance and profit of cloud infrastructure providers.

2.3. Distribute Resources in Physical Machines. Distributing the virtualized resources allocated in data center into the physical machines. In general, this step is defined as a Knapsack Problem or Bin Packing Problem. This step takes data center utilization, the migration overhead, and so forth, into account and needs further studies as well as analysis for virtualization.

Reference [4] proposes an approach for dynamic resource management in cloud which adapts a distributed architecture where resource management is decomposed into independent tasks, each of which is performed by Autonomous Node Agents through Multiple Criteria Decision Analysis using the PROMETHEE method. Thus [4] deals with C. Distribute resources in physical machines. Reference [5] proposes a resource management framework combining a utility-based dynamic VM provisioning manager and a dynamic VM placement manager. Both problems are modeled as Constraint Satisfaction Problems. And [5] deals with B. Resource brokering and provisioning and C. Distribute resources in physical machines. Reference [6] proposes an approach to managing infrastructure resources in PaaS by leveraging two adaptive control loops. The optimization loop improves the resource utilization of a cloud application via management functions provided by the corresponding middleware layers of PaaS. The allocation loop provides appropriate amounts of resources to/from the application system while guaranteeing its performance.

Based on [7] we know in such a Service-Oriented Architecture like cloud, the quality and reliability of the services become important aspects. However the demands of the service consumers vary significantly. From the service provider perspective, a balance needs to be made via a negotiation process since it is not possible to fulfill all consumer expectations. At the end of the negotiation process, provider and consumer commit to an agreement. In SOA terms, this agreement is referred to as a SLA (Service Level Agreement). The SLA serves as the foundation for the expected level of

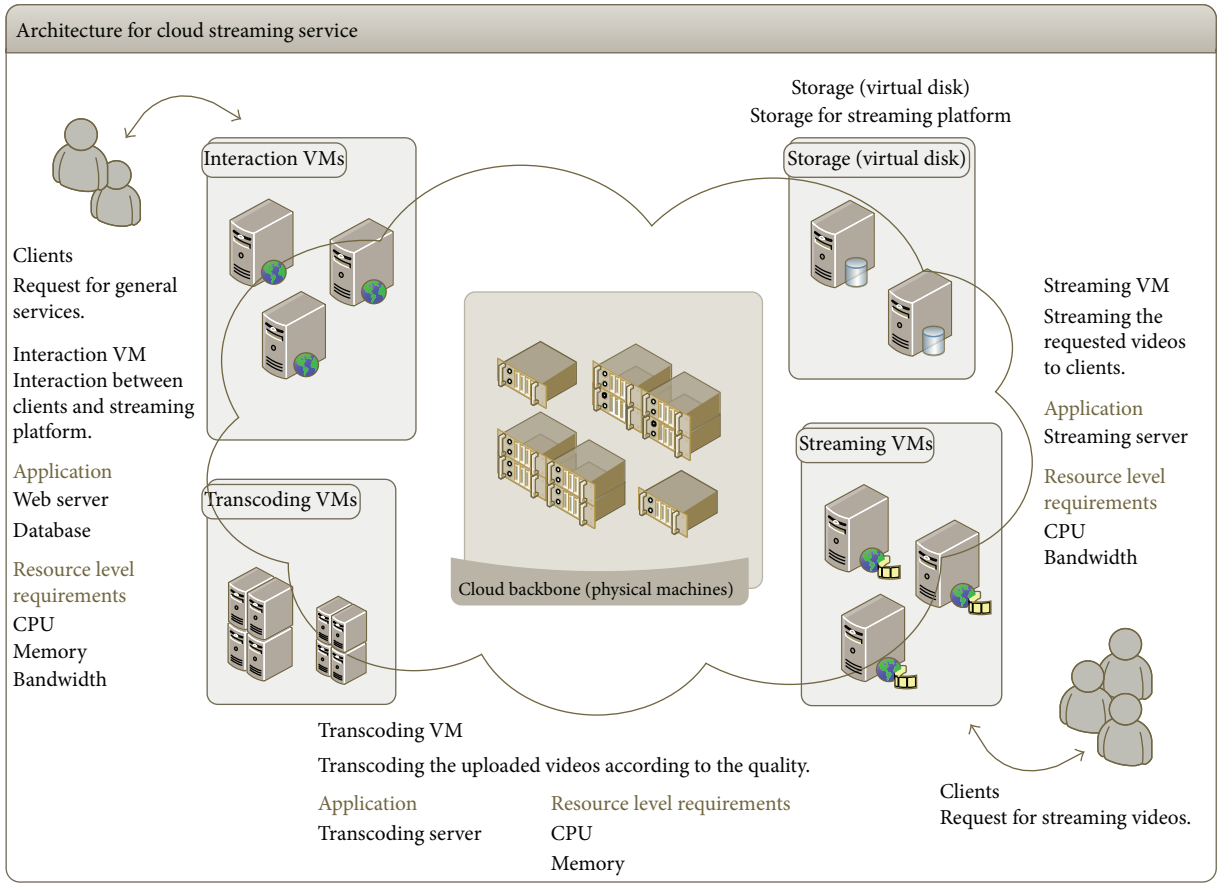


Figure 2: Cloud architecture for streaming service.

service between the consumer and the provider. In general, QoS requirements are part of an SLA.

SLA parameters are specified by metrics and usually included in a SLA parameters table. Metrics define how service parameters can be measured and are typically functions. There are at least two major types of metrics: (1) Resource metrics are retrieved directly from the provider resources and are used as is without further processing. (2) Composite metrics represent a combination of several resource metrics, calculated according to a special algorithm.

3. Cloud Streaming Service

3.1. Cloud Architecture for Streaming Service. As shown in Figure 2, the general system architecture of cloud computing environment for a typical streaming service, including but not limited to Video on Demand (VoD), live broadcasting, IPTV, and so forth, comprises numerous VMs. With the support of virtualization, the VMs can operate on the cloud backbone which consists of thousands of physical machines. These VMs can be categorized into four main types as follows.

(i) Interaction VMs (IVMs). They are responsible for the interaction between clients and streaming service platform. Those VMs host basic applications and services for platform operation, for example, web-based user interfaces and functions and database services. They also coordinate other types of VMs such as redirecting clients' requests for streaming videos to SVMs or queuing clients' requests for uploading videos, allocating enough storage, and assigning videos to TVMs. When considering the factors of resource requirements, bandwidth is the major concern for service capacity. Since many large websites use an in-memory datagrid for caching, applications of IVMs are mostly memory-intensive. Some applications are CPU-intensive due to their architecture.

(ii) Streaming VMs (SVMs). They are responsible for streaming the requested videos to clients. While receiving redirected requests from IVMs, the streaming applications hosted on SVMs process and packetize the source videos for specific format and protocol and then stream them to clients who issue the requests. The factors of resource requirements for SVMs are CPU and bandwidth.

(iii) Transcoding VMs (TVMs). Their duties are transcoding the uploaded videos according to the quality. Those VMs are responsible for transcoding source videos into varied quality videos with custom format settings. Transcoding is both CPU-intensive and memory-intensive. For many advanced encoders nowadays, multicore architectures with

high memory provision speed up the processes dramatically. The factors of resource requirements for TVMs are CPU and memory.

(iv) Storage. It is responsible for storage for streaming service platform. It usually consists of a large number of disk arrays. With the block virtualization and file virtualization, flexible management and optimization of storage utilization and server consolidation are achieved.

Note that those specific VM types are designed for VoD streaming services by the streaming service platform. Thus the mapping to VM instances of the IaaS providers as in Table 1 relies on further performance analysis by the streaming service platform. And those custom VM types may alter depending on the particular requirements of different services as well.

3.2. System Model and Assumption. In our proposed cloud streaming service model, there is a resource pool, P, comprising a set of physical machines (PM), $P = \{\mathrm{PM}_1, \mathrm{PM}_2, \ldots, \mathrm{PM}_n\}$. Assume all the physical machines are homogeneous in machine architecture and possess equivalent size, $\mathrm{PM}_{\mathrm{size}}$, for each considered resource factor. Through server virtualization/consolidation, a number of VMs can share a single PM, which increases utilization and in turn reduces the total number of PMs required. At any given time, we assume that n_{IVM}, n_{TVM}, and n_{SVM} are the amount of IVM, TVM, and SVM.

Based on the analysis of operations for streaming service, we outline three VM types as described above and an ideal resource mapping between the host application and their requirement for these types of VMs. Each VM type possesses varied price, PRI, and different capacity, CAP. The former is the cost of the streaming service platform to run a single VM for the duration, for example, \$0.075 per hour for $\mathrm{IVM}_{\mathrm{PRI}}$; the latter is the maximum capacity of one VM, for instance, 500 simultaneous connections for $\mathrm{IVM}_{\mathrm{CAP}}$. Generally speaking, there is a resource mapping table including PRI and CAP, for each type of VMs. This table can be obtained by modifying the pricing table from cloud infrastructure providers, such as Table 1.

At any given time, there is a set of requests R consisting of assorted tasks t which are classified by their VM host, namely, T_{IVM}, T_{TVM}, and T_{SVM}. T_{IVM} is a subset of R that consists of tasks for IVM; T_{SVM} and T_{TVM} are defined in the similar way; therefore $R = \{T_{\mathrm{IVM}}, T_{\mathrm{SVM}}, T_{\mathrm{TVM}}\}$. If there are i tasks of T_{IVM}, j tasks of T_{SVM}, and k tasks of T_{TVM}, then it must satisfy the following:

$$\begin{aligned} T_{\mathrm{IVM}} &= \left\{t_1^{\mathrm{IVM}}, t_2^{\mathrm{IVM}}, \ldots, t_i^{\mathrm{IVM}}\right\}, \\ T_{\mathrm{SVM}} &= \left\{t_1^{\mathrm{SVM}}, t_2^{\mathrm{SVM}}, \ldots, t_j^{\mathrm{SVM}}\right\}, \\ T_{\mathrm{TVM}} &= \left\{t_1^{\mathrm{TVM}}, t_2^{\mathrm{TVM}}, \ldots, t_k^{\mathrm{TVM}}\right\}, \\ &\text{s.t. } i + j + k = |R|. \end{aligned} \tag{1}$$

All request arrival rates follow different distributions. And each type of request possesses diverse execution time based on their characteristics as follows:

(i) Characteristics of T_{IVM} task: short-term execution, noncomputation-intensive, and strict response time.

(ii) Characteristics of T_{TVM} task: long-term execution, computation-intensive, and loose response time.

(iii) Characteristics of T_{SVM} task: execution time depending on media length, noncomputation-intensive, and strict response time.

Further numerical details about arrival rates and execution time distributions of requests will be discussed in Section 6. Also we list the key notations used in this paper in Notations. Note that it includes not only the notations mentioned in this section but also those defined in Section 4.

4. Criteria for Streaming Service

Although streaming is a well-developed application for the Internet, it can still be enhanced by leveraging the cloud technology such as parallel and distributed computing. Also, requests of a streaming service are quite varied in their resource requirements and execution time and this feature definitely fits in with the characteristics of on-demand and rapid elasticity in the cloud. These observations lead us to study streaming service as the service platform model. As illustrated in Figure 1, our resource broker adopts and transforms those criteria into the objective functions of the goal programming model. The criteria concerned by the streaming service and the cloud infrastructure provider, namely, *Service-oriented Criteria* and *Infrastructure-oriented Criteria*, are explored as follows.

4.1. Service-Oriented Criteria. *Service Utility*, including *Utilization* and *Availability*, is a set of criteria which measure the resource utilization and the accessibility and serviceability of the service platform. A resource is said to be critical to performance when it becomes overused or when its utilization is disproportionate to that of other components. On the other hand, availability means the percentage of time that the streaming service is available to process requests of clients.

(i) Utilization. It is assumed that VMs of the same type are load-balanced automatically. Thus utilization of IVM at any given time is $\mathrm{util}_{\mathrm{IVM}}$ as follows. Utilization of TVM and SVM are defined in the same way:

$$\mathrm{util}_{\mathrm{IVM}} = \begin{cases} 1, & \left|T_{\mathrm{IVM}}\right| \geq n_{\mathrm{IVM}} \times \mathrm{IVM}_{\mathrm{CAP}}, \\ \dfrac{\left|T_{\mathrm{IVM}}\right|}{n_{\mathrm{IVM}}}, & \left|T_{\mathrm{IVM}}\right| < n_{\mathrm{IVM}} \times \mathrm{IVM}_{\mathrm{CAP}}. \end{cases} \tag{2}$$

(ii) Availability. Availability of VM at any given time, $\mathrm{avail}_{\mathrm{IVM}}$, is defined in a similar way as *Utilization* and average availability of VM, $\mathrm{Avail}_{\mathrm{IVM}}$, during a time period, time_period, is as follows. Availability of TVM and SVM are defined in the same way:

$$\mathrm{Avail}_{\mathrm{IVM}} = \frac{\sum \mathrm{avail}_{\mathrm{IVM}}}{\mathrm{time\_period}}. \tag{3}$$

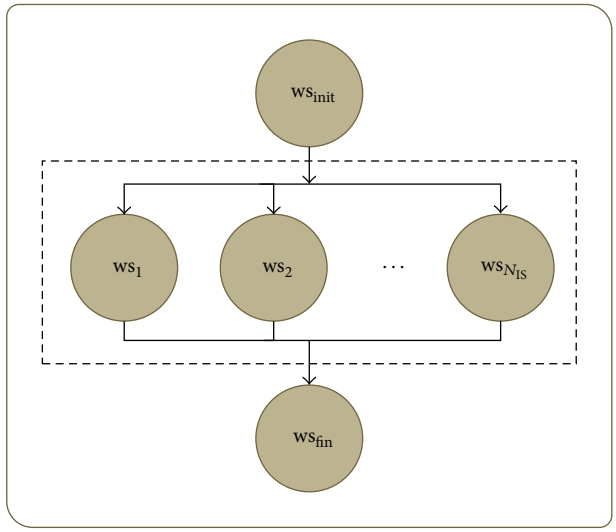


Figure 3: A composite web service.

QoS of Service Clients, which involves *response time*, *startup latency*, and *transcoding efficiency*, is a set of criteria which evaluate the Quality of Service (QoS) of the streaming service platform. Response time is concerned for IVM, while *startup latency* and *transcoding efficiency* are for SVM and TVM, respectively.

(iii) Response Time. Many large scale web sites take advantage of web services to boost their traffic. A single frontend Internet application may invoke many different web services. Such applications are called composite web services [8–10]. For example, browsing the streaming service platform by a registered member may invoke the web services of customization, recommendation, sorting, searching, and so forth.

Figure 3 is a model of a composite web service. After an initialization procedure, $\mathrm{WS}_{\mathrm{init}}$, N_{IS} web services are invoked and executed in parallel. Those parallel invocations of N_{IS} Web services will run on different IVMs and utilize the number of IVMs to speed up. The final procedure can only be carried after all N_{IS} web services have completed. Based on the analysis of the composite web service [8], we derive a function which can be measured in terms of utilization, the number of IVMs, and concurrent requests for IVM. Assume $\mathrm{ws}_{\mathrm{init}}$ and $\mathrm{ws}_{\mathrm{fin}}$ are the time of initial and final procedures in a composite web service and ws_i is the time it takes to execute web service i ($i = 1, \ldots, N_{\mathrm{IS}}$). Then the time it takes to execute N_{IS} web services that must synchronize after all completed is the maximum processing time among these N_{IS} web services. From [11–13], we know that based on queuing theory, as resource utilization increases, the response time per request rises dramatically. Reference [11] also derives a general formula to predict average response time:

$$\text{response } T = \frac{1}{1-p} \times \text{service } T, \tag{4}$$

where response T is average response time, service T is service time, and p is utilization. From Figure 3 and above we can formulate the response time rt of IVM at any given time as follows:

$$\mathrm{rt} = \mathrm{ws}_{\mathrm{init}} + \frac{1}{1-\mathrm{util}_{\mathrm{IVM}}} \left(\frac{N_{\mathrm{IS}}}{n_{\mathrm{IVM}}} \times \max_{1 \le i \le N_{\mathrm{IS}}} \{\mathrm{ws}_i\} \right) + \mathrm{ws}_{\mathrm{fin}}, \quad \text{s.t. } \frac{N_{\mathrm{IS}}}{n_{\mathrm{IVM}}} \ge 1, \ \mathrm{util}_{\mathrm{IVM}} \ne 1. \tag{5}$$

(iv) Startup Latency. As computers and network devices compress, encode, distribute, decompress, and render large amounts of data, buffers play a significant role in assuring the quality of digital media processing. In general, the larger the buffer, the better the end-user experience, but there is one main disadvantage of large buffers in a streaming scenario: Buffers cause delays or latencies. As shown in Figure 4, startup latency is the time a streaming service of SVM receives a request and starts media source processing, packetizing, and transmitting to the time that client fills up its buffer and starts playing. Here we focus on only buffering delay for streaming server. Thus network jitter, client download rate, and so forth, which can also affect startup latency, are not considered. According to the studies in [14–17], single-server streaming systems employ the server-push delivery model while client-pull architecture is more appropriate for multiserver streaming systems. Thus for our streaming service platform, client-pull architecture is assumed. In order to reduce startup latency, the basic idea is that clients need to fill up their buffers as fast as possible. It is vulnerable to network jitters if we tune down the buffer size of clients. Nevertheless, a client can issue multiple requests simultaneously for the video data segments and multiple streaming servers could be used for more throughput than a single server can provide.

The so-called server striping [15, 16] technique divides streaming video into fixed-size segments and distributes those segments over all SVMs. This helps us derive a function which can be estimated in terms of concurrent requests for SVM and the number of SVMs. We assume the policy of segment placement such as round-robin is adopted and the size of a streaming video is much larger than the size of a segment so that load imbalance due to uneven allocation between servers can be ignored. As illustrated in Figure 5, utilizing server striping technique to reduce the time needed to fill up the buffer, we assume the number of frames L which a buffer of client player contains is fixed and the average video processing frame rate for SVM is F. Then startup latency sl of SVM at any given time is defined as follows:

$$\mathrm{sl} = \frac{1}{n_{\mathrm{SVM}}} \times \frac{L}{F}, \quad \text{s.t. } \mathrm{avail}_{\mathrm{SVM}} = 1. \tag{6}$$

(v) Transcoding Efficiency. Based on the analysis of [18, 19], we assume that distributed transcoding architecture is used for the streaming service platform and we can exploit the cloud's elasticity to engage resources dynamically. Split and merge [20] technique enables us to split source video into segments

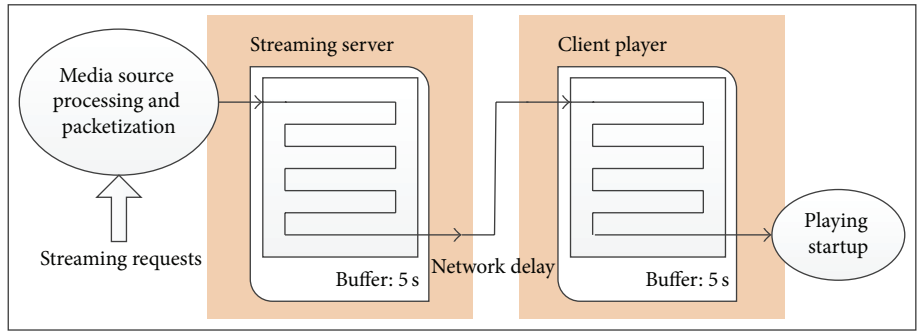


Figure 4: Startup latency in streaming.

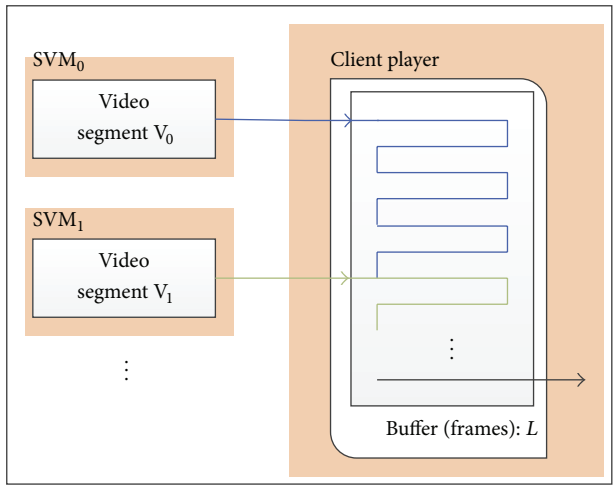


Figure 5: Reduce startup latency by server striping.

and distribute those segments over all TVMs to parallelize the video transcoding processes. Using this architecture, we assume the transcoding time of a single server is T_s and the transcoding time for a length of vt_time video of a cloud-based transcoding system of n_{TVM} TVMs is T_c. On system level, the gain of transcoding time from split and merge technique can be derived straightforwardly:

$$\frac{T_s}{T_c} \cong \frac{\text{vt\_time}}{\text{vt\_time} \times 1/n_{\text{TVM}}} \tag{7}$$

and as resource utilization increases, the transcoding time per request rises dramatically. Assume the degradation coefficient of utilization is α. Then transcoding efficiency te of TVM at any given time is

$$\text{te} = \alpha\left(1 - \text{util}_{\text{TVM}}\right) \times n_{\text{TVM}} \quad \text{s.t. } \text{util}_{\text{TVM}} < 1. \tag{8}$$

As mentioned in Section 1, those criteria with regard to *Service Utility* and *QoS of Service Clients* are defined as a SLA parameters table, which contains a set of QoS parameters and constraints as Table 2.

4.2. Infrastructure-Oriented Criteria. *Energy Consumption* and *profit performance* are criteria which measure the system energy consumption and profit of the cloud infrastructure providers.

Table 2: SLA parameters table.

SLA parameter	Constraint
Utilization	$\text{util}_{\text{IVM}} \geqq \text{constraint\_util}_{\text{IVM}}$ $\text{util}_{\text{SVM}} \geqq \text{constraint\_util}_{\text{SVM}}$ $\text{util}_{\text{TVM}} \geqq \text{constraint\_util}_{\text{TVM}}$
Availability	$\text{Avail}_{\text{IVM}} \geqq \text{constraint\_Avail}_{\text{IVM}}$ $\text{Avail}_{\text{SVM}} \geqq \text{constraint\_Avail}_{\text{SVM}}$ $\text{Avail}_{\text{TVM}} \geqq \text{constraint\_Avail}_{\text{TVM}}$
Response time	$\text{rt} \leqq \text{constraint\_rt}$
Startup latency	$\text{sl} \leqq \text{constraint\_sl}$
Transcoding efficiency	$\text{te} \geqq \text{constraint\_te}$

(i) Energy Consumption. Through server virtualization, a large number of users can share a single physical machine, which increases utilization and in turn reduces the total number of physical machines required. Reasonably cutting down the number of VMs can reduce the number of physical machines needed and achieve energy conservation. Since the services provided by IVM and SVM use a Software as a Service (SaaS) model, we can modify and employ the per-user energy consumption model for SaaS based on (9) in [21]:

$$P_{\text{sf}} = P_{\text{sf,PC}} + \frac{1.5P_{\text{sf,SR}}}{N_{\text{sf,SR}}} + 2B_d\frac{1.5P_{\text{SD}}}{B_{\text{SD}}} + AE_T. \tag{9}$$

The P_{sf} is per-user energy consumption of the SaaS service; $P_{\text{sf,PC}}$ is the power consumption of the user's terminal; $P_{\text{sf,SR}}$ is the power consumption of the server; $N_{\text{sf,SR}}$ is the number of users per server; B_d (bits) is the average size of a accessing file; P_{SD} is the power consumption of the hard disk arrays; B_{SD} is the capacity of the hard disk array; A is the transmission bit rate (bits per second); E_T is the per-bit energy consumption of transport in cloud computing. The multiplication by a factor of 2 in the third term accounts for the power requirements for redundancy in storage and the multiplication by a factor of 1.5 for second and third terms accounts for the energy consumption in cooling as well as other overheads.

Our main intention here is to derive functions to evaluate only the energy consumption of running physical machines for IaaS providers based on (9). Therefore we can eliminate the first term $P_{\mathrm{sf,PC}}$ since the end-user's energy consumption is not taken into account by service platform providers. The third term $2B_d(1.5P_{\mathrm{SD}}/B_{\mathrm{SD}})$ is ignored due to the fact that we do not consider storage issues in this paper. Also the last term AE_T which represents energy consumption of network transmission can be eliminated. From above and with our own notations, we can modify (9). Let $\mathrm{EC}_{\mathrm{IVM}}$ and $\mathrm{EC}_{\mathrm{SVM}}$ be the energy consumption of all physical machines which host all the IVMs and SVMs:

$$\begin{aligned}\mathrm{EC}_{\mathrm{IVM}} &= \frac{P_{\mathrm{sf}} \times \mathrm{IVM}_{\mathrm{CAP}}}{1.5} \times \frac{n_{\mathrm{IVM}}}{\mathrm{PM}_{\mathrm{size}}},\\ \mathrm{EC}_{\mathrm{SVM}} &= \frac{P_{\mathrm{sf}} \times \mathrm{SVM}_{\mathrm{CAP}}}{1.5} \times \frac{n_{\mathrm{SVM}}}{\mathrm{PM}_{\mathrm{size}}}.\end{aligned} \tag{10}$$

For services of TVM, per-user energy consumption model for Processing as a Service is adopted on the following basis (11) mentioned in [21]:

$$E_{\mathrm{ps}} = 40P_{\mathrm{ps,PC}} + 1.5NT_{\mathrm{ps,SR}}P_{\mathrm{ps,SR}} + 168AE_T. \tag{11}$$

The above per-user energy consumption (watt hours) E_{ps} is formulated as a function of the number of encodings per week N. $P_{\mathrm{ps,PC}}$ is the power consumption of the user's laptop; $T_{\mathrm{ps,SR}}$ is the average number of hours it takes to perform one encoding and $P_{\mathrm{ps,SR}}$ is the power consumption of the server.

The user's PC is used on average 40 h/week for common office tasks (factor of 40 in first term). A factor of 1.5 is included in the second term to account for the energy consumed to cool the computation servers as well as other over heads. In the third term, A is the per-user data rate (bits per second) between each user and the cloud, E_T is the per-bit energy consumption of transport, and the factor of 168 converts power consumption in transport to energy consumption per week (watt hours).

Here again the first term $40P_{\mathrm{ps,PC}}$ and the third term $168AE_T$ can both be eliminated due to the fact that end-user's energy consumption and energy consumption of network transmission are not taken into consideration.

Assume $\mathrm{EC}_{\mathrm{TVM}}$ is the energy consumption of all physical machines which host all the TVMs:

$$\mathrm{EC}_{\mathrm{TVM}} = \frac{E_{\mathrm{ps}} \times \mathrm{TVM}_{\mathrm{CAP}}}{1.5 \times N \times T_{\mathrm{ps,SR}}} \times \frac{n_{\mathrm{TVM}}}{\mathrm{PM}_{\mathrm{size}}} \tag{12}$$

and from above we can derive energy consumption ec at any given time:

$$\mathrm{ec} = \mathrm{EC}_{\mathrm{IVM}} + \mathrm{EC}_{\mathrm{SVM}} + \mathrm{EC}_{\mathrm{TVM}}. \tag{13}$$

(ii) Profit Performance. This criterion is evaluated by a function which can be measured in terms of the number of all types of VMs and their relative cost PRI. Therefore the profit performance pp can be formulated as follows:

$$\mathrm{pp} = n_{\mathrm{IVM}}\mathrm{IVM}_{\mathrm{PRI}} + n_{\mathrm{SVM}}\mathrm{SVM}_{\mathrm{PRI}} + n_{\mathrm{TVM}}\mathrm{TVM}_{\mathrm{PRI}}. \tag{14}$$

5. Goal Programming Model

5.1. MCDM and Goal Programming. Multicriteria Decision Making (MCDM) is a mathematical programming discipline under multiple objectives. It has emerged as a powerful tool to assist in the process of searching for decisions which best satisfy a multitude of conflicting objectives, and there are a number of distinct methodologies for multicriteria decision-making problems that exist. These methodologies can be categorized in a variety of ways, such as form of model (e.g., linear, nonlinear, and stochastic), characteristics of the decision space (e.g., finite or infinite), or solution process (e.g., prior specification of preferences or interactive). There are already many developed MCDM methods in use today and goal programming is one of the most popular and well-known techniques among them.

Goal programming (GP) is a multiobjective optimization technique which can cope with Multicriteria Decision Making problems. The essence of GP consists in the concept of satisfying of objectives. In fact, real-world problems invariably involve nondeterministic systems for which a variety of conflicting, noncommensurable objectives exist [22]. Due to the conflicts of objectives and the incompleteness of available information, it is almost impossible to build a reliable mathematical representation of the decision makers' preferences which has an optimal solution that optimizes all the objective functions. On the contrary, within such decision environment the decision makers try to achieve a set of goals which are represented by objective functions and constraints as closely as possible.

GP models can be classified into two major subsets [23], namely, weighted GP and preemptive/lexicographic GP. The general formulation of a weighted GP is given as follows:

$$\begin{aligned}\min \quad & z = \sum_{i=1}^{k} \left(u_i n_i + v_i p_i\right)\\ \text{s.t.} \quad & f_i(x) + n_i - p_i = b_i,\\ & i = 1 \cdots Q,\end{aligned} \tag{15}$$

where $f_i(x)$ is a linear or nonlinear objective function of x and b_i is the constraint for that objective. The unwanted negative and positive deviations n_i and p_i are assigned weights according to their relative importance to the decision maker and minimized.

In preemptive GP, the deviational variables are assigned into a number of priority levels and minimized in a preemptive way. The formulation of a preemptive GP is given as follows:

$$\begin{aligned}\min \quad & a = \left\{g_1(n,p), g_2(n,p), \ldots, g_L(n,p)\right\}\\ \text{s.t.} \quad & f_i(x) + n_i - p_i = b_i,\\ & i = 1 \cdots Q.\end{aligned} \tag{16}$$

In this model, α is an ordered vector of these L priority levels and g_L is a function of the deviation variables associated with the objectives or constraints at priority level L.

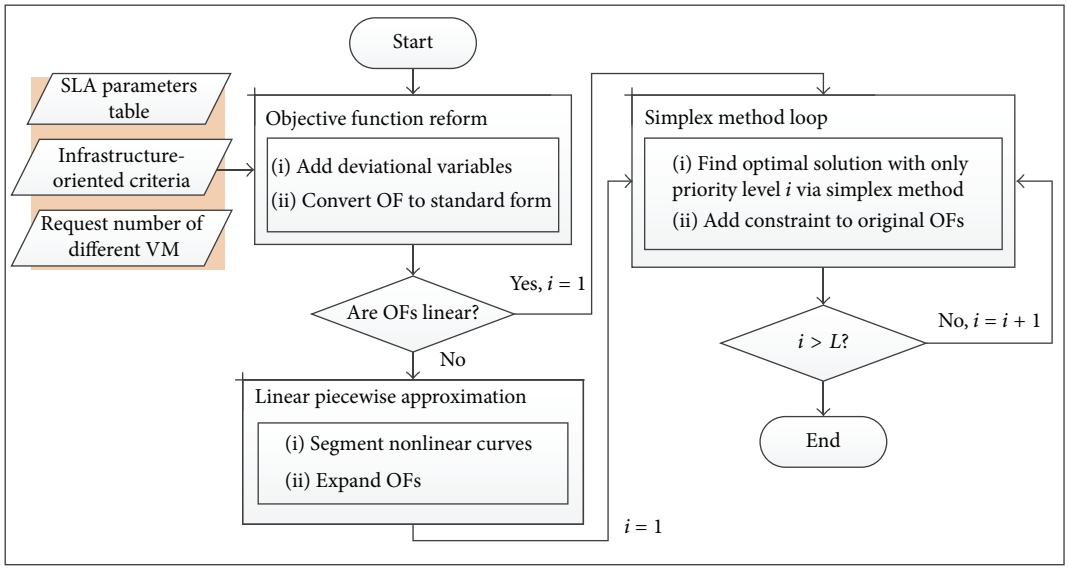


Figure 6: The goal programming procedure.

5.2. Problem Formulation. The resource allocation/provisioning problem in our autonomous resource broker is to judge the quantity of IVMs, TVMs, and SVMs while deliberating upon those conflicting criteria. Let n_{IVM}, n_{TVM}, and n_{SVM} be the number of IVM, TVM, and SVM. The determination of n_{IVM}, n_{TVM}, and n_{SVM} is a Multiple Criteria Decision Making problem for satisfying contrary goals of service-oriented and infrastructure-oriented. This problem can be formulated and solved based on preemptive Goal Programming. Assume p_i is the unwanted positive deviation which numerically exceeds the ith goal and let n_i be the unwanted negative deviation which falls short of the ith goal. We can express this problem as follows:

$$\min \quad \{(p_4 + p_5 + n_6), (n_1 + n_2 + n_3), (p_7 + n_8)\} \tag{17}$$

$$\text{subject to} \quad \text{util}_{\text{IVM}} + n_1 = \text{constraint\_util}_{\text{IVM}} \tag{18}$$

$$\text{util}_{\text{SVM}} + n_2 = \text{constraint\_util}_{\text{SVM}} \tag{19}$$

$$\text{util}_{\text{TVM}} + n_3 = \text{constraint\_util}_{\text{TVM}} \tag{20}$$

$$\text{rt} - p_4 = \text{constraint\_rt} \tag{21}$$

$$\text{sl} - p_5 = \text{constraint\_sl} \tag{22}$$

$$\text{te} + n_6 = \text{constraint\_te} \tag{23}$$

$$\text{ec} - p_7 = \text{constraint\_ec} \tag{24}$$

$$\text{pp} + n_8 = \text{constraint\_pp}. \tag{25}$$

The equations above from (18) to (25) are objective functions of the goal programming model. These equations are formulations of related criteria derived in Section 4 and each of these equations has a specific constraint defined by the streaming service platform. Equations (18), (19), and (20) stand for *Utilization* of IVM, SVM, and TVM, and (21), (22), and (23) are regarded as *response time*, *startup latency*, and *transcoding efficiency*, respectively. And the last two equations, (24) and (25), are viewed as *energy consumption* and *profit performance*.

From the perspective of the streaming service platform, *Service-oriented Criteria* are always more important than *Infrastructure-oriented Criteria*. And criteria for *QoS of Service Clients* are considered first compared to criteria for *Service Utility*. Thus in (17), we first minimize n_4, n_5, and p_6 which represent the unwanted deviations for *response time*, *startup latency*, and *transcoding efficiency*. Then we minimize p_1, p_2, and p_3 that stand for the unwanted deviations for *Utilization* of IVM, SVM, and TVM. Lastly we minimize n_7 and p_8 which count as the unwanted deviations for *energy consumption* and *profit performance*.

5.3. Goal Programming Approach. In general, for each priority level, our goal programming approach identifies basic feasible solutions and refines them iteratively using Simplex algorithm to achieve the best possible compromise solution. The detailed procedure is illustrated in Figure 6. There are three main processes, namely, *Objective Function Reform*, *Linear Piecewise Approximation*, and *Simplex Method Loop*.

(i) Objective Function Reform. In this procedure, first it imports mathematical formulation and constraints from SLA Parameters table (derived from service-oriented criteria) and infrastructure-oriented criteria as the objective functions of goal programming model. Then the deviational variables are defined and added to all the objective functions since it is not known whether a given solution will undersatisfy or oversatisfy the goals or constraints set by criteria.

We seek to minimize the nonachievement of the goals or constraints by minimizing specific deviation variables. Before the simplex method can be used to solve, all the objective functions must be converted into an equivalent functions

Table 3: Object function reform.

Goal or constraint	Objective function	Deviational variables to be minimized
$f_i(x) \leqq b_i$	$f_i(x) + n_i - p_i = b_i$	p_i
$f_i(x) \geqq b_i$	$f_i(x) + n_i - p_i = b_i$	n_i
$f_i(x) = b_i$	$f_i(x) + n_i - p_i = b_i$	$n_i + p_i$

in which all constraints are equations and all variables are nonnegative. The way of adding the deviational variables and converting objective functions into standard form can be summarized in Table 3. The priority of each goal is also adjusted according to the relative importance of service-oriented criteria and infrastructure-oriented criteria, as well as the request number for IVM, SVM, and TVM.

(ii) Linear Piecewise Approximation. After reforming the objective functions, we should verify if these objective functions are linear. As with many real-world problems the functional representations of the objective functions were mostly nonlinear.

Reference [24] gives a method of modeling any monotonically increasing or decreasing, nonlinear, and discontinuous function while remaining within the GP format. Generally speaking, nonlinear objective function curves can be segmented by a piecewise linear approximation according to [25] and the resulting straight-line segments can then be modeled using penalty or reverse penalty function methods which expands the original objective functions as described in [24]. Apparently, the more the number of segments used, the greater the accuracy in modeling the objective functions. Nevertheless using more segments also increases the size of the resulting GP model.

(iii) Simplex Method Loop. After objective function reform and linear piecewise approximation, a standard linear GP model is built. Thus we can construct a sequential goal programming code to solve our model in this procedure. The mechanism is described as follows:

(a) Let i be the priority level under consideration and L is the total priority level. Set $i = 1$.

(b) Solve priority level i only. That is to say, minimize $\alpha_i = g_i(n, p)$, subject to only the goals or constraints associated with i. Such a problem is equivalent to a traditional single-objective model which can be solved via simplex method. Let the optimal solution to this problem be given as α_i^*; namely, α_i^* is the optimal solution for $g_i(n, p)$.

(c) Set $i = i + 1$; if $i > L$, go to (d). Add constraint for the optimal solution of previous priority levels; namely, $g_i(n, p) = \alpha_j^*$, $j = 1, \ldots, i - 1$, to original goal programming model for the next priority level. Then go to (b).

(d) The end of the loop and the solution vector, associated with the last single objective model solved, is the optimal vector for the original goal programming model.

The simplex method/algorithm is a well-known algorithm of solving linear programming problems. It is created by George Dantzig in 1947 and used for planning and decision-making in large-scale enterprises. Based on chapter 4 of [26], the general procedure of simplex algorithm is given as follows.

Step 1. Convert the linear programming problem to the standard form.

Step 2. Obtain a BFS (basic feasible solution) from the standard form.

Step 3. Determine whether the current BFS is optimal. If current BFS is not optimal, go to Step 4, else the procedure reaches the end.

Step 4. If the current BFS is not optimal, then find a new BFS with a better objective function value. Then go to Step 3.

The concept of this algorithm is derived from the name "simplex." Simplex is a generalization of the notion of a triangle or tetrahedron to arbitrary dimension and the geometrical interpretation of the behavior of simplex algorithm is that its search procedure follows a simplex and essentially starts from some initial corner point and then follows a path along the edges of the feasible region towards an optimal corner point. Note that all the intermediate corner points visited are improving (more precisely, not worsening) the objective function.

6. Performance Evaluation

In this section, six criteria are considered and evaluated, namely, Utilization of three types of VMs, *response time*, *startup latency*, *transcoding efficiency*, *energy consumption*, and *profit performance*. In order to judge the performance of the proposed goal programming approach, we compared it with a utility-based model which is adopted in [27]. Its concept has been used in microeconomic theory. Here we modify it with our notations and develop it as utility-based model. Let utilization and price be the measured utility function of the utility-based model.

We use MATLAB R2010a as our simulation tool and the settings of parameters are shown in Table 4. According to probability theory and statistics, we simulate the demand for T_{IVM}, T_{SVM}, and T_{TVM} per hour using Poisson distribution with various values of λ. The "off-peak" zones are 5~8. The "normal" zones are 0~2 and 15~18. The "peak" zones are 11~13 and 20~23. For T_{IVM}, there are "off-peak" hours with $\lambda = 200$, "normal" hours with $\lambda = 300$, and "peak" hours with $\lambda = 500$. For T_{SVM}, there are "off-peak" hours with $\lambda = 250$, "normal" hours with $\lambda = 350$, and "peak" hours with $\lambda = 550$. For T_{TVM}, there are "off-peak" hours with $\lambda = 300$, "normal" hours with $\lambda = 350$, and "peak" hours with $\lambda = 400$.

For *response time*, we assume that a composite web service is composed of N_{IS} independent web services and $N_{\mathrm{IS}} = 15$.

Table 4: Simulation parameters.

Parameter	Value
Simulation time	24 hours (mean of 300 days)
IVM_{CAP}	100 requests
SVM_{CAP}	50 requests
TVM_{CAP}	20 requests
IVM_{PRI}	\$0.8 per hour
SVM_{PRI}	\$0.8 per hour
TVM_{PRI}	\$0.9 per hour
ws_{init}	0.8 ms
ws_{fin}	0.7 ms
N_{IS}	15
L	600 frames
F	60 fps
α	1.7
P_{sf}	10 kWh
E_{ps}	12 kWh
PM_{size}	10

Also ws_{init} and ws_{fin} are the time of initial and final procedures in a composite web service. For *Startup Latency*, the number of frames L which a buffer of client player contains is 600 frames and the average video processing frame rate for SVM is $F = 60$ frames per second. For *transcoding efficiency*, the degradation coefficient of utilization $\alpha = 1.7$. For energy consumption, we assume that the per-user energy consumption for IVM and SVM service is $P_{sf} = 10$ kWh and the per-user energy consumption for TVM is $E_{ps} = 12$ kWh. Also all the physical machines possess equivalent size, $PM_{size} = 10$, for IVM, SVM, and TVM.

Based on the utility model of [27], the utility concept has been used in microeconomic theory. Here we modify it with our notations and develop it as utility-based pricing model. Let utilization and price be the measured utility function of the utility-based pricing model:

$$\begin{aligned} \max \quad & U = \sum \left(\text{util}_{IVM}, \text{util}_{SVM}, \text{util}_{TVM}\right) \\ \text{subject to} \quad & \min \text{PRI} \\ & = \sum \left(n_{IVM}\text{IVM}_{PRI} + n_{SVM}\text{SVM}_{PRI} + n_{TVM}\text{TVM}_{PRI}\right). \end{aligned} \tag{26}$$

We use this utility-based pricing model in comparison with our proposed goal programming approach in the simulation.

6.1. Utilization. The platform utilization of IVM, SVM, and TVM during the simulation hours is shown in Figure 7. The horizontal axis represents simulation time and the vertical axis stands for resource utilization of platform. Note that we set the following: constraint_$\text{util}_{IVM} \geqq 60\%$, constraint_$\text{util}_{SVM} \geqq 60\%$, and constraint_$\text{util}_{TVM} \geqq 60\%$.

For utility-based model, the slope of platform utilization of IVM, SVM, and TVM during the simulation hours is more gradual than GP. We can see that util_{IVM} of utility-based model is always greater than 80%, util_{SVM} of utility-based model is always greater than 90%, and util_{TVM} of utility-based model is always greater than 95%. However for GP, most of the time the curves of platform utilization of IVM and SVM are lower than utility-based model and they even drop obviously during the off-peak and the normal period. Note that util_{IVM} of GP and util_{SVM} of GP are lower than 60% during the off-peak period but higher than 60% during the normal and the peak period. Basically, utilization of utility-based model is better than GP, especially during the off-peak period. This is due to the fact that priority of QoS of Service Clients criteria is higher than Service Utility criteria in goal programming approach. While GP is worse than utility-based model in utilization, GP still achieves the goals of constraint_util_{IVM}, constraint_util_{SVM}, and constraint_util_{TVM} mostly.

6.2. Response Time. Figure 8 shows the response Time of IVM. The horizontal axis represents simulation time and the vertical axis stands for response time in millisecond. Note that we set the following constraints for GP: constraint_rt $\leqq$ 3.5 milliseconds.

Utility-based model has a steep slope compared with GP. And the response time of utility-based model rises to almost 6 and 4.5 milliseconds during normal and off-peak period while the response time of GP is around 3.5 milliseconds (slightly higher during off-peak and normal but lower during peak period) most of the time. Obviously the response time of GP is always better than utility-based model and achieves the goal of constraint_rt mostly. Compared with Figure 7, we can derive that, during off-peak and normal period, GP makes a trade-off between utilization of IVM and response time and the latter is prior to the former, whereas utility-based model always consider utilization only.

6.3. Startup Latency. Figure 9 shows the *Startup Latency* of SVM during the simulation hours. The horizontal axis represents simulation time and the vertical axis stands for startup latency in seconds. Note that we set the following constraints for GP: constraint_sl $\leqq$ 1 second.

Similarly, utility-based model has a steep slope compared with GP. The startup latency of utility-based model rises to almost 2 and higher than 1 seconds during normal and off-peak period while the startup latency of GP is around 0.9 milliseconds (again slightly higher during off-peak and normal but lower during peak period) most of the time. We can conclude that the startup latency of GP is always better than utility-based model and achieves the goal of constraint_sl mostly. And compared with Figure 7, we can conclude that, during off-peak and normal period, GP makes a trade-off between utilization of SVM and startup latency and the latter is prior to the former, whereas utility-based model always considers utilization only.

6.4. Transcoding Efficiency. The *transcoding efficiency* of TVM during the simulation hours is shown in Figure 10. The horizontal axis represents simulation time and the vertical axis stands for transcoding efficiency of TVM. Note that we set the following constraints for GP: constraint_te $\geqq 4$.

Both utility-based model and GP have gradual slopes. The transcoding efficiency of utility-based model is at almost 2.5

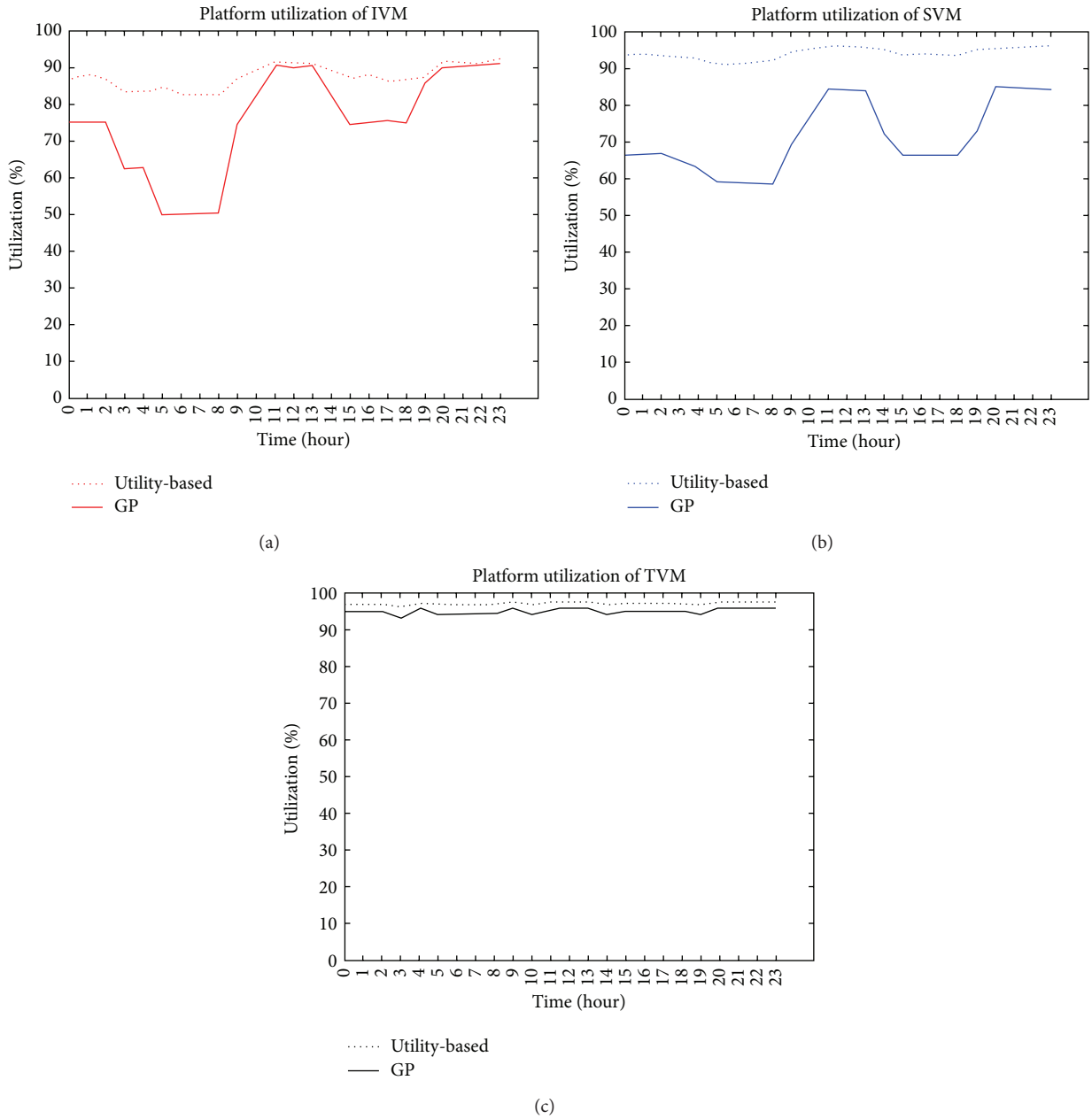


Figure 7: (a) Resource utilization of streaming service platform for IVM. (b) Resource utilization of streaming service platform for SVM. (c) Resource utilization of streaming service platform for TVM.

while the transcoding efficiency of GP is above 4. Clearly the transcoding efficiency of GP is always better than utility-based model and can achieve the goal of constraint_te all the time.

6.5. Energy Consumption and Profit Performance. Here we show the results of energy consumption and profit performance in Figures 11 and 12. The horizontal axis represents simulation time and the vertical axis stands for energy consumption in kWh and profit performance in $.

The energy consumption of utility-based model is always lower than GP which indicates that the amount of IVM, SVM, and TVM of utility-based model is less than GP and so is the number of the host physical machines. That is to say, GP always tends to allocate more resources in order to maintain *QoS of Service Clients*. On the other hand, more resources allocated of the service platform means more profit for the cloud infrastructure provider. Utility-based model is not considered what QoS level should be achieved or SLA should be followed. Thus, utility-based model just tries to reach the needs of physical machine. On the contrary, GP in order to conform Quality of Service and SLA for clients, it is necessary to acquire more resources to achieve the goal. Naturally, for the service provider, more resources required means more profit; for the client, GP provides a better and stable service experience.

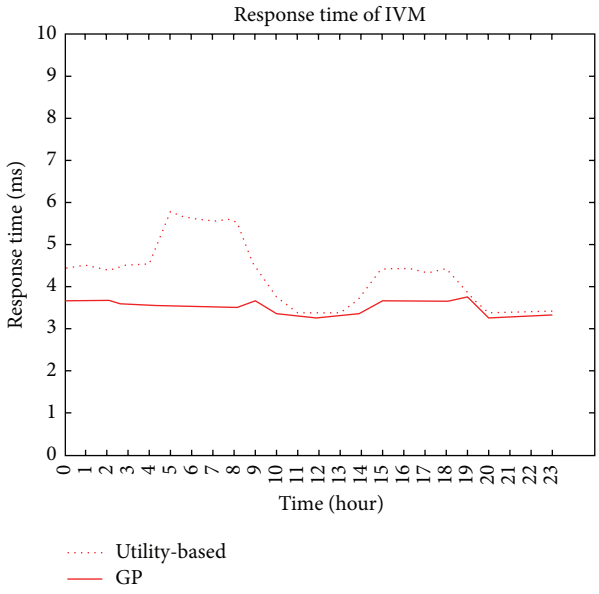


Figure 8: Response time of IVM.

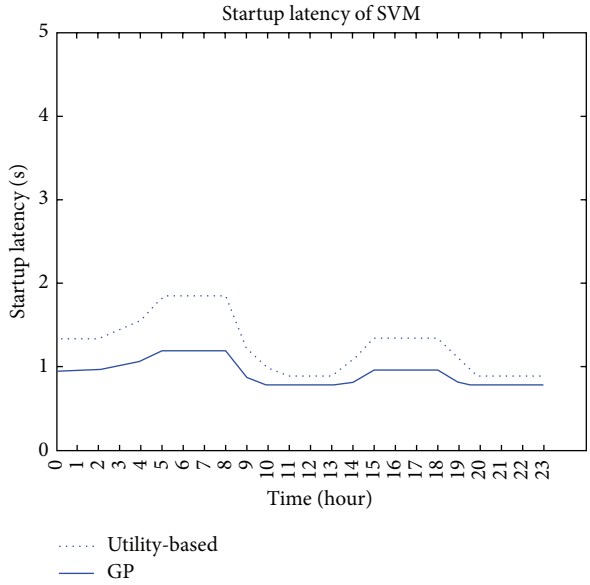


Figure 9: Startup latency of SVM.

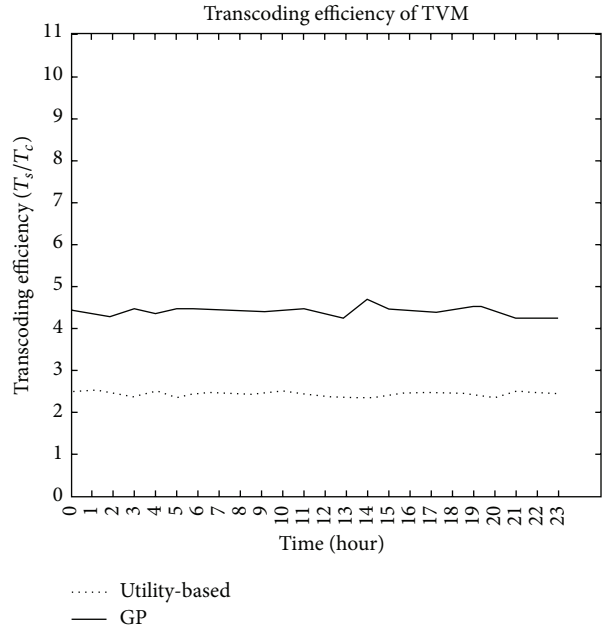


Figure 10: Transcoding efficiency of TVM. Energy consumption and profit performance.

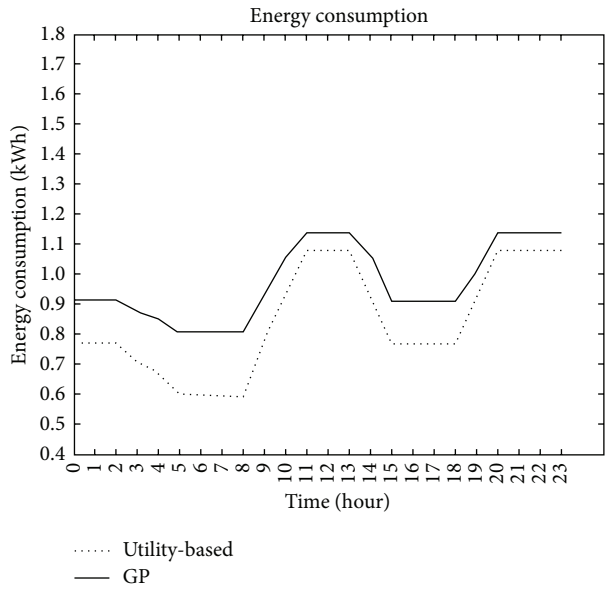


Figure 11: Energy consumption.

7. Conclusion

In this paper, we have addressed the problem of resource brokering between the streaming service platform and the cloud infrastructure provider considering both service-oriented and infrastructure-oriented criteria. An effective resource provisioning algorithm for an autonomous resource broker in the cloud is proposed and the core mechanism of this broker is obtained by formulating and solving a goal programming model. This resource broker manages the provisioning of cloud resources between the streaming service platform and the cloud infrastructure provider based on not only the number of incoming requests for the various services of the streaming platform but also both the service-oriented criteria which are in the form of SLA and infrastructure-oriented criteria. Also, a cloud architecture for the streaming service is proposed and extensive analysis is performed for related criteria. The simulation results show that the proposed approach makes an effective trade-off between these conflicting criteria commendably and achieves our goals of QoS.

For the future work, we hope that the approaches proposed in our work can further be practiced and applied to a real cloud environment, for example, being implemented as a

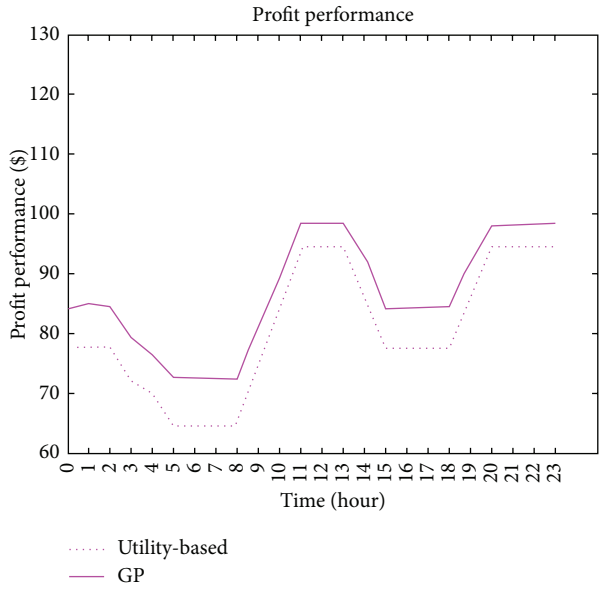


Figure 12: Profit performance.

decision making system used to make decision plans for efficiently provisioning resources considering multiple criteria for both cloud infrastructure providers and their customers. Also for resource brokering, there are still many performance issues that need further analysis in more complex deployment models like hybrid cloud.

Notations

P: Resource pool comprising a set of physical machines
PM_{size}: The number of VMs that a single PM can host
IVM: Interaction virtual machine
SVM: Streaming virtual machine
TVM: Transcoding virtual machine
n_{IVM}: The amount of IVM
n_{SVM}: The amount of SVM
n_{TVM}: The amount of TVM
IVM_{PRI}: The cost to run a single IVM for a duration
SVM_{PRI}: The cost to run a single SVM for a duration
TVM_{PRI}: The cost to run a single TVM for a duration
IVM_{CAP}: Maximum capacity of a single IVM for a duration
SVM_{CAP}: Maximum capacity of a single SVM for a duration
TVM_{CAP}: Maximum capacity of a single TVM for a duration
R: A set of platform clients' requests
t: The tasks which are ran by their specific VM hosts
T_{IVM}: A subset of R that consists of tasks for IVM
T_{SVM}: A subset of R that consists of tasks for SVM
T_{TVM}: A subset of R that consists of tasks for TVM
$util_{IVM}$: Average resource utilization of all IVMs
$util_{SVM}$: Average resource utilization of all SVMs
$util_{TVM}$: Average resource utilization of all TVMs
$Avail_{IVM}$: Average availability of IVM during a time period
$Avail_{SVM}$: Average availability of SVM during a time period
$Avail_{TVM}$: Average availability of TVM during a time period
rt: Response time of IVM services
sl: Startup latency of SVM services
te: Transcoding efficiency of TVM services
EC_{IVM}: Energy consumption of all IVMs
EC_{SVM}: Energy consumption of all SVMs
EC_{TVM}: Energy consumption of all TVMs
ec: Overall energy consumption of these VMs
pp: Profit performance evaluated by the allocated VMs.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported by the Ministry of Science and Technology under Grant MOST 103-2221-E-309-004.

References

[1] P. Mell and T. Grance, *The NIST Definition of Cloud Computing (Draft)*, NIST Special Publication 800-145, National Institute of Standards and Technology, 2011.

[2] P. Barham, B. Dragovic, K. Fraser et al., "Xen and the art of virtualization," in *Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP '03)*, pp. 164–177, October 2003.

[3] Amazon EC2, http://aws.amazon.com/ec2/.

[4] Y. O. Yazir, C. Matthews, R. Farahbod et al., "Dynamic resource allocation in computing clouds using distributed Multiple Criteria Decision Analysis," in *Proceedings of the 3rd IEEE International Conference on Cloud Computing (CLOUD '10)*, pp. 91–98, July 2010.

[5] H. N. Van, F. D. Tran, and J.-M. Menaud, "Performance and power management for cloud infrastructures," in *Proceedings of the 3rd IEEE International Conference on Cloud Computing*, pp. 329–336, July 2010.

[6] Y. Zhang, G. Huang, X. Liu, and H. Mei, "Integrating resource consumption and allocation for infrastructure resources on-demand," in *Proceedings of the 3rd IEEE International Conference on Cloud Computing (CLOUD '10)*, pp. 75–82, July 2010.

[7] P. Patel, A. Ranabahu, and A. Sheth, "Service level agreement in cloud computing," in *Proceedings of the Workshop on Best Practices in Cloud Computing: Implementation and Operational*

Implications for the Cloud at ACM International Conference on Object-Oriented Programming, Systems, Languages, and Applications, October 2009.

[8] D. A. Menascé, "QoS issues in web services," *IEEE Internet Computing*, vol. 6, no. 6, pp. 72–75, 2002.

[9] D. A. Menascé, "Response-time analysis of composite web services," *IEEE Internet Computing*, vol. 8, no. 1, pp. 90–92, 2004.

[10] D. A. Menascé, "Composing web services: a QoS view," *IEEE Internet Computing*, vol. 8, no. 6, pp. 88–90, 2004.

[11] J. Gray and A. Reuter, *Transaction Processing: Concepts and Techniques*, Morgan Kaufmann Publishers, San Francisco, Calif, USA, 1993.

[12] C. Devlin, "SaaS Capacity Planning: Transaction Cost Analysis Revisited," 2008, http://msdn.microsoft.com/en-us/library/cc261632.aspx.

[13] H. Al-Hilali, D. Guimbellot, and M. Oslake, *Capacity Model for Internet Transactions*, 1999, http://research.microsoft.com/pubs/69700/tr-99-18.doc.

[14] S. S. Rao, H. M. Vin, and A. Tarafdar, "Comparative evaluation of server-push and client-pull architectures for multimedia servers," in *Proceedings of the 6th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '96)*, Zushi, Japan, April 1996.

[15] J. Y. B. Lee, "Parallel video servers: A tutorial," *IEEE Multimedia*, vol. 5, no. 2, pp. 20–28, 1998.

[16] J. Y. B. Lee and P. C. Wong, "Performance analysis of a pull-based parallel video server," *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, no. 12, pp. 1217–1231, 2000.

[17] J. Y. B. Lee, "Buffer management and dimensioning for a pull-based parallel video server," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 4, pp. 485–496, 2001.

[18] Z. Tian, J. Xue, W. Hu, T. Xu, and N. Zheng, "High performance cluster-based transcoder," in *Proceedings of the International Conference on Computer Application and System Modeling (ICCASM '10)*, pp. V248–V252, October 2010.

[19] Y. Sambe, S. Watanabe, D. Yu, T. Nakamura, and N. Wakamiya, "Distributed video transcoding and its application to grid delivery," in *Proceedings of the 9th Asia-Pacific Conference on Communications (APCC '03)*, vol. 1, pp. 98–102, Penang, Malaysia, 2003.

[20] K. Breitman, M. Endler, R. Pereira, and M. Azambuja, "When TV dies, will it go to the cloud?" *Computer*, vol. 43, no. 4, Article ID 5445174, pp. 81–83, 2010.

[21] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker, "Green cloud computing: Balancing energy in processing, storage, and transport," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 149–167, 2010.

[22] J. P. Ignizio, "A review of goal programming: a tool for multiobjective analysis," *Journal of the Operational Research Society*, vol. 29, no. 11, pp. 1109–1119, 1978.

[23] M. Tamiz, D. Jones, and C. Romero, "Goal programming for decision making: an overview of the current state-of-the-art," *European Journal of Operational Research*, vol. 111, no. 3, pp. 569–581, 1998.

[24] D. F. Jones and M. Tamiz, "Expanding the flexibility of goal programming via preference modelling techniques," *Omega*, vol. 23, no. 1, pp. 41–48, 1995.

[25] H. P. Williams, *Model Building in Mathematical Programming*, John Wiley & Sons, New York, NY, USA, 1978.

[26] W. L. Winston, *Operations Research: Applications and Algorithms*, Duxbury Press, 3rd edition, 1997.

[27] J.-T. Park, J.-W. Baek, and J. W.-K. Hong, "Management of service level agreements for multimedia internet service using a utility model," *IEEE Communications Magazine*, vol. 39, no. 5, pp. 100–106, 2001.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 408921, 7 pages
http://dx.doi.org/10.1155/2015/408921

Research Article

A Replication-Based Mechanism for Fault Tolerance in MapReduce Framework

Yang Liu and Wei Wei

College of Information Science and Engineering, Henan University of Technology, Zhengzhou 450001, China

Correspondence should be addressed to Wei Wei; nsyncw@126.com

Received 20 October 2014; Accepted 31 January 2015

Academic Editor: Hui-Huang Hsu

MapReduce is a programming model and an associated implementation for processing and generating large data sets with a parallel, distributed algorithm on a cluster. In cloud environment, node and task failure are no longer accidental but a common feature of large-scale systems. Current rescheduling-based fault tolerance method in MapReduce framework failed to fully consider the location of distributed data and the computation and storage overhead of rescheduling failure tasks. Thus, a single node failure will increase the completion time dramatically. In this paper, a replication-based mechanism is proposed, which takes both task and node failure into consideration. Experimental results show that, compared with default mechanism in Hadoop, our mechanism can significantly improve the performance at failure time, with more than 30% decreasing in execution time.

1. Introduction

MapReduce is an emerging programming paradigm that has gained more and more popularity due to its ability to support complex tasks execution in a scalable way [1, 2]. MapReduce is used for processing large data sets by parallelizing the processing on a large number of distributed nodes. Data is stored in splits that are processed by separate tasks. Processing is done in two phases: map and reduce. A MapReduce application is implemented in terms of two functions that correspond to these two phases. A map function processes input data expressed in terms of key-value pairs and produces an output also in the form of key-value pairs. A reduce function picks the output of the map functions and produces results. It is shown that many applications can be implemented using this programming model [1].

This popularity is also shown by the appearance of open-source implementations, like Hadoop that is now extensively adopted by Yahoo and many other companies [3]. Hadoop [4] is an open-source software framework implemented using Java and is designed to support data-intensive applications executed on large distributed systems. It is a project of the Apache Software Foundation and is a very popular software tool due, in part, to it being open-source. Yahoo! has contributed to about 80% of the main core of Hadoop, but many other large technology organizations have used or are currently using Hadoop, such as, Facebook, Twitter, LinkedIn, and others [5]. The Hadoop framework is comprised of many different projects, but two of the main ones are the Hadoop Distributed File System (HDFS) and MapReduce. Both the initial input and the final output of a Hadoop MapReduce application are normally stored in HDFS [3], which is similar to the Google File System [6].

However, in large-scale distributed computing environment such as cloud environment, node and task failure are no longer *accidental* but a common feature. Research discovered that failure has a significant impact on system performance in large-scale systems [4]. Every year in a cluster, 1% to 5% hard disks will be scrapped, up to 20 racks and 3 routers will go down, and servers will go down at least twice with 2% to 4% scrap rate each year. It shows that failure also occurs daily even in a distributed system with up to ten thousand super reliable servers (MTBF of 30 years) [5]. For the cloud environment consisting of a large number of inexpensive computers, node and task failure become a more frequent and wide spread problem, which must be handled by some

effective fault tolerance method. It is difficult to achieve 100% fault tolerance because there are many physical circumstances that just can not be planned for, but the goal of fault tolerance is to plan for all common failures [7]. In managing fault tolerance it is important to eliminate Single Points of Failure (SPOF), which are single elements of the system, that when they fail, they can bring down the whole system [8].

As a result, fault tolerance is as important in the design of the original MapReduce as in Hadoop. Specifically, a MapReduce job is a unit of work that consists of the input data, a map and a reduce function, and configuration information. Hadoop breaks the input data in splits. Each split is processed by a map task, which Hadoop prefers to run on one of the nodes where the split is stored (HDFS replicates the splits automatically for fault tolerance). Map tasks write their output to local disk, which is not fault tolerance. However, if the output is lost, as when the machine crashes, the map task is simply executed again on another computing node. The outputs of all map tasks are then merged and sorted by an operation called shuffle. This kind of rescheduling is inefficient and can be improved in many ways.

Since the MapReduce-based programs generate a lot of intermediate data, which is critical for completing the job, this paper views failover of intermediate data as a necessary component of MapReduce framework, specifically targeting and minimizing the effect of tasks and nodes failure on performance metrics such as job completion time. We propose new design techniques for a new fault tolerance mechanism, implement these techniques within Hadoop, and experimentally evaluate the prototype system.

2. Related Work

MapReduce is a programming model and an associated implementation for processing and generating big data [9]. It is initially designed for parallel processing of big data using mass cheap server clusters and putting scalability and system availability on the prior position. Within Google Company, more than 20 PB of data is produced every day and 400 PB every month. Yahoo adopts Hadoop, an open-source MapReduce framework. Facebook uses it to process data and generate reports, while Amazon Company uses elastic MapReduce framework for large amount of data-intensive tasks [10]. MapReduce has obvious advantage over other programming models like MPI, and a single task failure does not affect the execution of other tasks because of the independence between mapper and reducer task. It has drawn a lot of attention for its benefits in simple programming, data distribution, and fault tolerance, which has been widely used in many areas, like data mining and machine learning.

Google Company pointed out that, in a cloud environment with averagely 268 nodes, each MapReduce job is accompanied with failure of five nodes [11]. On the other hand, large amount of data is usually accompanied with data inconsistency or even data loss, and incorrect data record will result in task failure or even failure of the entire job. MapReduce uses a rescheduling-based fault tolerance mechanism to ensure the correct execution of the failed task. Since the rescheduling failed to fully consider the location of distributed data, in the scenario of node failure, all the completed tasks on the failed node will start over, which shows severely low efficiency. If the failure detection timeout in Hadoop is set to 10 minutes (the default value), a single failure will cause at least 50% increase in completion time [12]. If each input split contains one bad record, the entire MapReduce job will have a 100% runtime overhead, which is not acceptable for those users with rigorous SLA requirements. So it clearly shows the need for effective algorithms that can reduce delays caused by these failures [13].

In [14], tests show that, in seven types of cluster with different MTBF (Mean Time between Failures), MapReduce job with three replicas can achieve better performance than that with one replica, because more replicas can reduce the chances of data migration when rescheduling jobs at failure time. Reference [15] discussed an alternative fault tolerance scheme, the state-based Stream MapReduce (SMR), which is suitable for handling continuous data streaming applications with real-time requirements, such as financial and stock data. The key feature is low-overhead deterministic execution which reduces the amount of persistently stored information. Reference [16] proposed a method to replicate intermediate data to the reducer, but this method will produce a large number of I/O operations, consume a lot of network bandwidth, and only support recovery for single node failure. Reference [17] proposed a method to improve performance of fault tolerance by replicating data copies. Reference [18] presented an intelligent scheduling system for web service, which considers both the requirements of different service requests and the circumstances of the computing infrastructure which consists of various resources. Reference [19] described the priority of fairness, efficiency, and the balance between benefit and fairness, respectively, and then recompiled the CloudSim and simulated three task scheduling algorithms on the basis of extended CloudSim, respectively.

We proposed a replication-based mechanism for fault tolerance in MapReduce framework, which can significantly reduce the average completion time of jobs. Unlike traditional fault tolerance mechanism, it will reschedule tasks on the failed node to another available node without starting over again but reconstruct intermediate results quickly from the checkpoint file. The preliminary experiments show that, under a failure condition, it outperforms default mechanism with more than 30% increasing in performance and incurs only up to 7% overhead.

3. Algorithms

3.1. MapReduce Programming Model. In MapReduce programming model, the calculation process is decomposed into two main phases, namely, the mapper stage and reducer stage. For one piece of input data, the reducer stage only starts when the mapper stage is completed. A MapReduce job includes M mapper tasks and R reducer tasks. In mapper stage, multiple mapper tasks run in parallel, and one mapper task will read an input split and perform a mapper function,

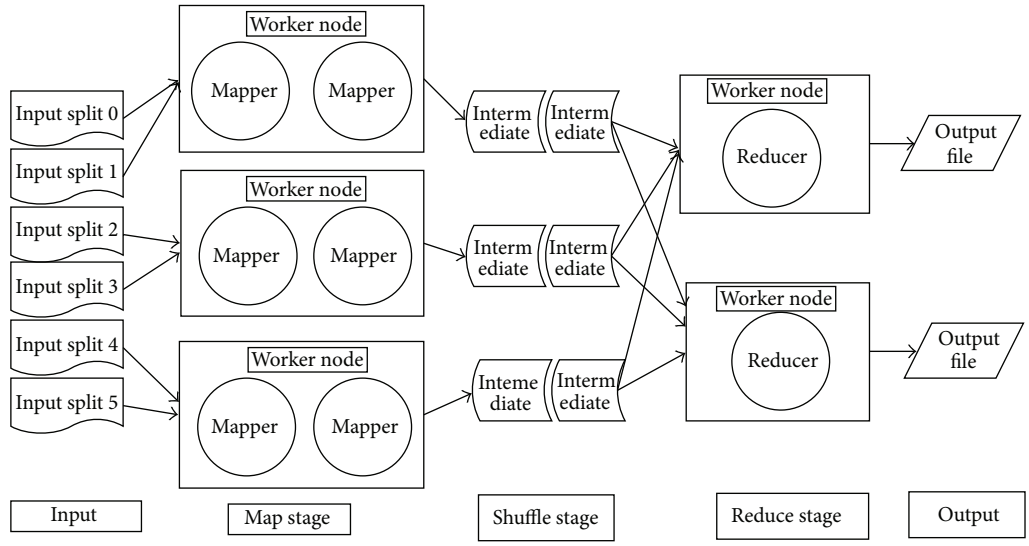


Figure 1: Execution process of MapReduce model.

where mapper tasks are independent of each other. Mapper tasks will produce a large number of intermediate results in local storage. Before reducer function is called, the system will classify the generated intermediate result and shuffle result with the same key to reducers. A reducer task will execute a reduce function and generates an output file, and eventually a MapReduce job will generate R output files which can be merged to get the final result. When programming, developers need to write a mapper and a reducer function:

$$\begin{aligned} &\text{map: } (\text{input\_data}) \longrightarrow \left\{\left(\text{key}_j, \text{value}_j\right) \mid j = 1, \ldots, k\right\} \\ &\text{reduce: } \left(\text{key}, \left[\text{value}_1, \ldots, \text{value}_m\right]\right) \longrightarrow \left(\text{key}, \text{final\_value}\right). \end{aligned} \tag{1}$$

The MapReduce model is shown in Figure 1.

Node and task failure are prone to happen during a MapReduce job execution process. When a node fails, MapReduce will move all mapper tasks on the failed node to available nodes. This kind of rescheduling method is simple but often introduces a lot of time cost; thus for users with high responsiveness requirements, its negative influence is not acceptable.

3.2. Improved Rescheduling Algorithm. In the paper, a replication-based mechanism in MapReduce framework is proposed, which uses a checkpoint-based active replication method to provide better performance. Both node and task failures are considered and the introduced delay is significantly decreased; thus the overall performance is improved.

In our mechanism, two kinds of files are introduced, namely, the local checkpoint file and the global index file. These files are created before the execution of one mapper task. The local checkpoint file is responsible for recording the progress of current task, avoiding reexecution from the beginning in case of task failure. If local task failure happened, one node can continue failed task using information in local checkpoint file. And the global index file is responsible for recording the characteristics of the current job, thereby helping in reconstructing the intermediate results across nodes to reduce reexecution time. The global index file can be implemented as a checkpoint file saved in HDFS and can be accessible in case of node failure.

The algorithm is divided into two parts: one part works on master node and the other works on worker nodes. In addition, as the master node is important, it is necessary to maintain multiple fully consistent "hot backup," to ensure seamless migration when a fault occurs. The two parts of algorithm are shown as follows.

The Algorithm on Master Node

(1) The master node preassigns mapper and reducer task to different worker nodes.

(2) Choose K replicas for each worker node.

(3) Wait for results from all worker nodes.

  (a) If all results are received, merge these results and mark job as completed.

  (b) Or go to 3 and keep waiting.

(4) Periodically send probing packets to all worker nodes.

  (a) If all worker nodes respond, then go to 4 and keep probing.

  (b) Or if one node does not respond in given time interval, then mark the node as failed.

    (i) Get the worker ID and all unfinished tasks on the node.

    (ii) Put all unfinished mapper tasks into global queue, and reschedule them on available replica nodes.

(iii) If there are failed tasks on the failed node, then reschedule these tasks on replica nodes which have intermediate results, without reexecuting these mapper tasks.

(c) If one node has finished all tasks, then reassign other unexecuted tasks to the node.

The Algorithm on Worker Node

(1) Check the type of the given task.

(a) If it is a mapper task, check whether it is a new task or a reexecution of failed task.

(i) If it is a new task, initialize and execute it.

(ii) If it is a reexecution of local failed task, then get its progress from the local checkpoint file and continue execution.

(iii) If it is a reexecution of the failed task from other nodes, then read global index file for the task and rapidly reconstruct intermediate result using information in global index file.

(b) If it is a reducer task, check whether it is a new task, a reexecution of the failed task, or unexecuted task from other nodes.

(i) If it is a new task, initialize and execute it.

(ii) If it is a reexecution of the failed task from other nodes, read intermediate data from the local disk and execute it.

(iii) If it is a new assigned unexecuted task from other nodes, then read intermediate data from a given worker node and execute it.

(2) Create a local checkpoint file and a global index file for the given mapper task.

(3) Start the mapper task.

(a) When memory buffer of the mapper node is full, dump intermediate data into local files. After dumping finished, record the position of the input stream and mapper ID (Position, mapper_ID) into local checkpoint file.

(b) According to the location distribution of input key-value pairs which contributes to output key-value pairs, two different strategies are employed to record these distributions in the global index file.

(i) For input key-value pairs producing output, record their position (T1, offset) into global index file, which means only pairs in these offsets need to be processed in reexecution. Here T1 means this is the type 1 record.

(ii) Or, for input pairs which give no output, record the range as (T2, offset1, offset2), which means the input pairs between offset1 and offset2 have no output and can be skipped in reexecution. T2 means this is the type 2 record.

(4) When a mapper task finished, shuffle and send the intermediate results to corresponding reducer nodes. Copy the intermediate data needed by local reducer task to replica nodes, then notify the completion of mapper task to master node, and delete the local checkpoint file and the global index file.

In our algorithm, when a task failure occurs, the corresponding node simply reads the checkpoint file saved in the local disk, restore task status to the checkpoint, and reload the intermediate results generated before failure, so reexecution is avoided.

When a node failure occurs, the scheduler on the master node is responsible for rescheduling the interrupted mapper tasks to available replica nodes, which can quickly construct the intermediate results of failed tasks using the global index file, to reduce the reexecution time greatly.

Note if tasks and nodes failed before checkpoint, the progress will continue from the recent checkpoint. And if the action of saving checkpoint failed, the progress will start from the recent checkpoint again. The simple failover strategy is of low cost and is effective in distributed computing environment. The frequency of saving checkpoint should be carefully chosen: a high frequency will provide lower cost for failover and higher checkpoint saving cost for normal running, while a low frequency is in the opposite case. After careful adjusting, the frequency value in our experiment is set to one checkpoint per 105 key-value pairs.

If node failure happens in the reducer stage, the tasks on the failed node are rescheduled to any available replica nodes. The needed intermediate results have been copied to the replica node when mapper tasks finished; thus there is no need to repeat the mapper tasks on the failed node, so that overall completion time of the MapReduce job is greatly reduced.

3.3. System Analysis. For a given mapper task m, the question is, for a given input range in an input split, how to properly place the kind of records (T_1 or T_2) in global index file, to minimize storage overhead. For the given input range, set the total number of key-value pairs as N_m and the number of output key-value pairs as N'_m. The size of type 1 and type 2 records is L_1 and L_2:

$$L_2 = 2L_1. \tag{2}$$

Set $S_{m,1}$ and $S_{m,2}$ as the storage overhead in type 1 and type 2 record in the input range. Set V_m as the count of input subranges which give no output, and the O_ith input key-value pair produces the ith output key-value pairs. Consider

$$V_m = \sum_{i=2}^{N'_m} \begin{cases} 0, & O_i - O_{i-1} = 1 \\ 1, & O_i - O_{i-1} > 1. \end{cases} \tag{3}$$

We have

$$S_{m,1} = L_1 N'_m$$
$$S_{m,2} = L_2 V_m = 2L_1 V_m. \quad (4)$$

Set R_m as the decision we make; when $R_m = 1$, we store type 1 record to global index file, or if $R_m = 2$, we use type 2 record. Then we can decide which kind of record to use according to equation as follows:

$$R_m = \begin{cases} 1, & N'_m < 2V_m \\ 2, & N'_m \geq 2V_m. \end{cases} \quad (5)$$

4. Experimental Results

We validate our algorithm on Hadoop cluster, which is implemented as a patch to Hadoop. The comparison is measured from the performance and the overhead aspect. The performance of the algorithm is evaluated by delay. The implementation of the algorithm is based on Hadoop 0.20.1, Java 1.6, and HDFS file system with data block size of 256 MB. The underlying infrastructure is a 20-node HP blade cluster and each node is equipped with quad-core Xeon 2.6 GHz CPU, 8 G memory, 320 G hard drive, and two pieces of Gigabit NICs. Each node is configured to hold four Xen virtual machines; thus it has 80 virtual nodes. And we schedule 40 nodes for Hadoop cluster with default fault tolerance mechanism and 40 nodes for cluster with our mechanism. For each cluster, one node is deployed as the master node and the remaining 39 nodes are deployed as the worker node. A single worker node can simultaneously run two mapper tasks and one reducer task. In the experiment we run a typical filter job, which is to find out certain entries from huge amounts of data. This kind of job is computationally intensive and has less intermediate results. We use 1.2 million English web pages with an average page size of 1 MB for test. By adjusting the split size, one mapper handles an average of approximately 120 M input data, and each node is assigned with an average of about 250 mapper tasks.

According to the distribution of query words in input data, there are three kinds of MapReduce jobs used in the experiment: the aggregated job, the sparse job, and the mixed job. In an aggregated job, the locations of query words are gathered in the target data; in a sparse job, the locations of the query words are more dispersed; in a hybrid job, the above two situations coexist.

The processing time of MapReduce job is usually influenced by factors as below: (1) the size of the data set, where each mapper task will deal with a subset of data (split); (2) the computation performance of worker nodes; (3) MapReduce job type; (4) the number of bad records in each split, where each record will lead to restarting of task; (5) the number of failed worker nodes. By adjusting these factors, the effectiveness and the overhead of given fault tolerance algorithms can be evaluated and compared in the same criterion.

In scenario with only task failure, the comparison of job completion time of two mechanisms is shown in Figure 2, with our mechanism marked as RFM (replication-based fault tolerance mechanism). The x-axis is the task error probability in the form of error tasks number per 100 tasks, with the y-axis being the total completion time. There is no upper limit on the errors number of each mapper task. It can be observed that, along with the increasing of probability of errors, the execution time of the job with REF is significantly decreased compared to that of default mechanism.

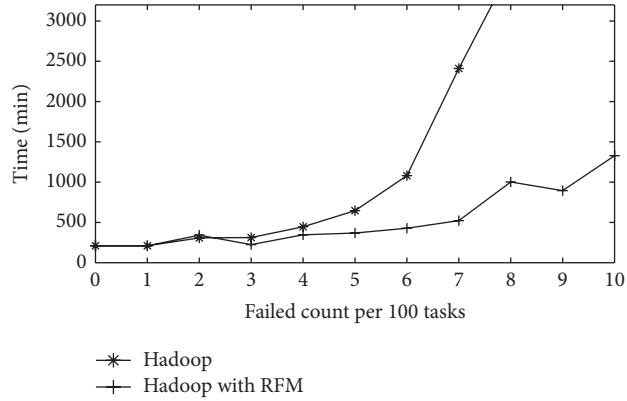


Figure 2: Comparison of execution time with task failure.

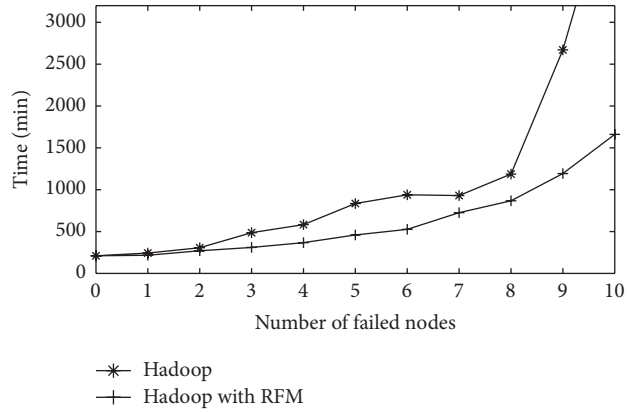


Figure 3: Comparison of execution time with node failure.

In scenario with only node failure, the comparison of MapReduce job completion times is shown in Figure 3. The x-axis is the number of failed nodes, with the y-axis being the total completion time. It can be observed that, along with more failed nodes, our mechanism can significantly decrease the reexecution time. It is because the default mechanism of Hadoop will start all failed task from the beginning on replica nodes, while our mechanism gets more tasks finished in the same period of time by continuing the mapper tasks quickly and efficiently.

In scenario with both task and node failure, the comparison of job completion times is shown in Figure 4. The x-axis is the probability of task failure and node failure, in the form of failed tasks/nodes count per 100 tasks/nodes, with y-axis as the total completion time. It is shown that, under the joint influence of task and node failure, the relation between the job execution time and the failure probability is nearly linear

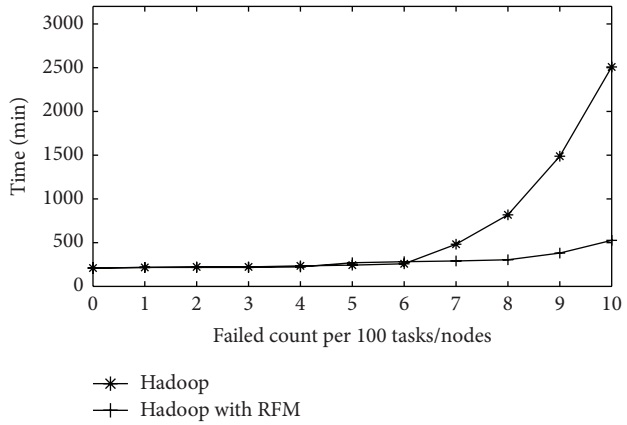


Figure 4: Comparison of execution time with task and node failure.

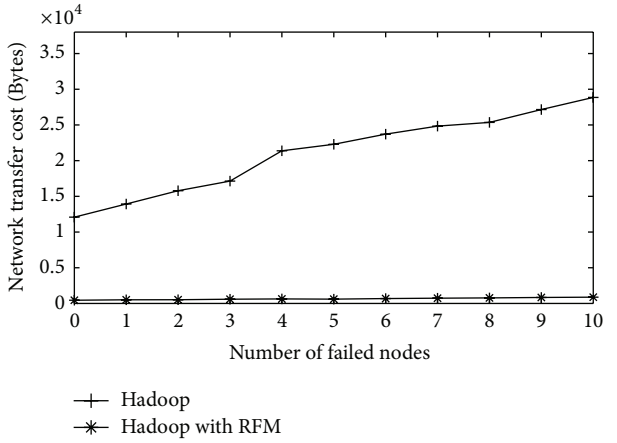


Figure 5: Comparison of average network overhead with node failure.

in our mechanism, and the overhead is mainly introduced by task migration and restoring.

Moreover, the overhead introduced by our mechanism is low. In case of task failure, the network overhead can be ignored since there is no extra network transferring. In case of node failure, the extra network overhead of our mechanism is mainly from the active replication of the global index file. Figure 5 shows the network overhead which is introduced by node failure. The x-axis is the number of failure nodes, and the y-axis is the average network overhead. It is shown that, compared with the network overhead of default mechanism in Hadoop, the network overhead of our mechanism is significantly low.

In the task failure scenario, the storage overhead is the size of the local checkpoint file, which is negligible due to the limited position information stored in it. Figure 6 shows the comparison of storage overhead of three different types of MapReduce job, under the scenario of 10 failed nodes. It is shown that the increased storage overhead of our mechanism is mainly from global index file, which is low compared with original intermediate results.

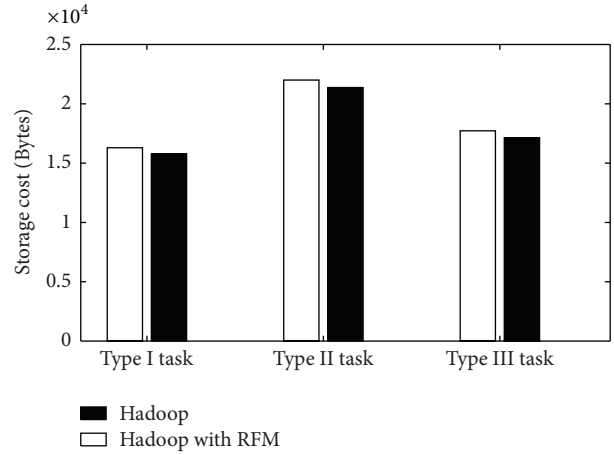


Figure 6: Comparison of average storage overhead with node failure.

5. Conclusions

The paper proposed a replication-based mechanism for fault tolerance in MapReduce framework, which is fully implemented and tested on Hadoop. Experimental results show the runtime performance can be improved by more than 30% in Hadoop; thus our mechanism is suitable for multiple types of MapReduce job and can greatly reduce the overall completion time under the condition of task and node failures.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This paper is supported by the National Natural and Science Foundation of China (nos. 61003052, 61103007), Natural Science Research Plan of the Education Department of Henan Province (nos. 2010A520008, 13A413001, and 14A520018), Henan Provincial Key Scientific and Technological Plan (no. 102102210025), Program for New Century Excellent Talents of Ministry of Education of China (no. NCET-12-0692), Natural Science Research Plan of Henan University of Technology (2014JCYJ04), and Doctor Foundation of Henan University of Technology (nos. 2012BS011, 2013BS003).

References

[1] J. Dean and S. Ghemawat, "MapReduce: simplified data processingon large clusters," in *Proceedings of the 6th Symposiumon Operating Systems Design & Implementation*, vol. 3, pp. 102–111, 2004.

[2] U. Hoelzle and L. A. Barroso, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*, Morgan and Claypool Publishers, 2009.

[3] T. White, *Hadoop: The Definitive Guide*, O'Reilly, 2009.

[4] Apache Hadoop, http://hadoop.apache.org.

[5] D. Borthakur, The Hadoop Distributed File System: Architecture and Design, http://hadoop.apache.org/docs/r0.18.0/hdfs_design.pdf.

[6] S. Ghemawat, H. Gobioff, and S. T. Leung, "The Google file system," *SIGOPS Operating Systems Review*, vol. 37, no. 5, pp. 29–43, 2003.

[7] B. Selic, "Fault tolerance techniques for distributed systems," http://zh.scribd.com/doc/37243421/Fault-Tolerance-Techniques-for-Distributed-Systems.

[8] F. Wang, J. Qiu, J. Yang, B. Dong, X. Li, and Y. Li, "Hadoop high availability through metadata replication," in *Proceedings of the 1st International Workshop on Cloud Data Management (CloudDB '09)*, pp. 37–44, ACM, November 2009.

[9] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[10] Amazon Elastic Mapreduce, http://aws.amazon.com/elasticmapreduce/.

[11] J. Dean, "Experiences with MapReduce, an abstraction for large-scale computation," in *Proceedings of the Internet Conference on Parallel Architectures and Computation Techniques (PACT '06)*, vol. 8, pp. 5–10, Seattle, Wash, USA, 2006.

[12] S. Y. Ko, I. Hoque, B. Cho, and I. Gupta, "On availability of intermediatedata in cloud computations," in *Proceedings of the The USENIX Workshop onHot Topics in Operating Systems (HotOS '09)*, vol. 5, pp. 32–38, Ascona, Switzerland, 2009.

[13] J.-A. Quiané-Ruiz, C. Pinkel, J. Schad, and J. Dittrich, "RAFTing MapReduce: fast recovery on the RAFT," in *Proceedings of the IEEE 27th International Conference on Data Engineering (ICDE '11)*, vol. 3, pp. 589–600, IEEE, Hannover, Germany, April 2011.

[14] J. Hui, Q. Kan, S. Xian-He, and L. Ying, "Performance under failures of mapreduce applications," in *IEEE/ACM International Symposium on Cluster Cloud and Grid Computing*, pp. 608–609, Newport Beach, Calif, USA, 2011.

[15] A. Martin, T. Knauth, S. Creutz et al., "Low-overhead fault tolerance for high-throughput data processing systems," in *Proceedings of the 31st International Conference on Distributed Computing Systems (ICDCS '11)*, pp. 689–699, Minneapolis, Minn, USA, June 2011.

[16] S. Y. Ko, I. Hoque, B. Cho, and I. Gupta, "Making cloud intermediate data fault-tolerant," in *Proceedings of the 1st ACM Symposium on Cloud Computing*, vol. 1, pp. 181–192, Indianapolis, Ind, USA, June 2010.

[17] Q. Zheng, "Improving mapreduce fault tolerance in the cloud," in *Proceedings of the IEEE International Symposium on Parallel Distributed Processing Workshops and Phd Forum (IPDPSW '10)*, vol. 1, pp. 1–6, New York, NY, USA, 2010.

[18] J. Liu, X. G. Luo, B. N. Li, X. M. Zhang, and F. Zhang, "An intelligent job scheduling system for web service in cloud computing," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 11, no. 12, pp. 2956–2961, 2013.

[19] S. Hong, S.-p. Chen, J. Chen, and G. Kai, "Research and simulation of task scheduling algorithm in cloud computing," *Telkomnika*, vol. 11, no. 11, pp. 1923–1931, 2013.

Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 617840, 14 pages
http://dx.doi.org/10.1155/2015/617840

Research Article

A Mobile Cloud Computing Framework Integrating Multilevel Encoding for Performance Monitoring in Telerehabilitation

Saiyi Li, Hai Trieu Pham, M. Sajeewani Karunarathne, Yee Siong Lee, Samitha W. Ekanayake, and Pubudu N. Pathirana

School of Engineering, Deakin University, Australia

Correspondence should be addressed to Saiyi Li; saiyi@deakin.edu.au

Received 27 March 2015; Revised 10 June 2015; Accepted 11 June 2015

Academic Editor: Bao Rong Chang

Recent years have witnessed a surge in telerehabilitation and remote healthcare systems blessed by the emerging low-cost wearable devices to monitor biological and biokinematic aspects of human beings. Although such telerehabilitation systems utilise cloud computing features and provide automatic biofeedback and performance evaluation, there are demands for overall optimisation to enable these systems to operate with low battery consumption and low computational power and even with weak or no network connections. This paper proposes a novel multilevel data encoding scheme satisfying these requirements in mobile cloud computing applications, particularly in the field of telerehabilitation. We introduce architecture for telerehabilitation platform utilising the proposed encoding scheme integrated with various types of sensors. The platform is usable not only for patients to experience telerehabilitation services but also for therapists to acquire essential support from analysis oriented decision support system (AODSS) for more thorough analysis and making further decisions on treatment.

1. Introduction

Resource deficiencies in the healthcare industry to cater the rapid changes in demography entail the development of new rehabilitative practices [1] enabling therapists to provide their services to remote communities. Since the first article using the term "telerehabilitation" published in 1998 [2], this new rehabilitation practice paradigm has been widely studied for its capabilities in reducing patients' commute to hospital and saving time and cost for both clinicians and patients [3]. Telerehabilitation can be defined as a set of instruments and protocols aimed at providing access to rehabilitation services for patients at geographically distant locations [4].

Carignan and Krebs [5] mentioned that one of the challenges in developing a telerehabilitation system is to reduce the delay in the data communication and processing, especially where real-time biofeedback is required. As is observed, the major delay in the telerehabilitation is caused by the low bandwidth of the Internet, large amount of data, and the low computational power of the computer used to process captured data. Therefore, the delay resulted from the increase in users can be partially addressed by increasing the computational power of the computer hosting the server side of the telerehabilitation system. Furthermore, the increase in users is associated with a significant increase in captured data from various sensors, which requires the flexibility in capacity of the database. Additionally, applying pervasive computing in healthcare, such as telerehabilitation, is becoming popular in the recent years [6, 7], which leads to a challenge on how to integrate existing technologies to a telerehabilitation system.

With recent developments in ICT, mobile cloud computing (MCC) [8] has emerged to bring cloud computing [1] power to mobile devices. MCC [8] can be considered the ideal candidate to accommodate majority of the above criteria.

Firstly MCC inherently provides architecture to support various mobile devices, such as smartphones, tablets, and laptops, which means that users of telerehabilitation services can readily access the service with less restrictions on the time and location.

Secondly, similar to generic cloud computing, MCC also provides an elastic approach to support the increasing number of users. One example is Amazon Web Services (AWS) [9], which enables users to expand the computation power and data storage space easily. More

importantly, users deploying their services in cloud only pay for the resources they use, which is also called "pay-as-you-go" (PAYG), thereby reducing the cost significantly compared to traditional approach to deploy services.

Thirdly, the computation power in MCC enables performing calculation fast enough to offer patients real-time biofeedback, which is critical in telerehabilitation [10] to improve the effectiveness of telerehabilitation session [11–13]. Because of the limited computational power and battery resources, mobile devices are unlikely to handle this task for a long time. Therefore, offloading the computation tasks to the cloud reduces the load in mobile devices and enables telerehabilitation users to access the service as long as possible.

Although MCC has provided a well-designed platform to establish telerehabilitation services, there exist some gaps to be solved for this particular application. One of the most critical issues is how to reduce the energy consumption in mobile devices so that patients are able to access telerehabilitation services as long as possible. In addition, another open question is how to design a telerehabilitation platform that caters for various types of sensors to collect sufficient data in a short period so that automatic assessment and analysis of patients' performance can be performed in the therapeutic procedures as an ongoing treatment scheme.

This paper primarily seeks the answers for the previously mentioned questions. The main contribution of the paper is a multilevel data encoding scheme for optimisation of computational offloading and data transferring. Although a vast array of studies discussed about strategies to optimise the computational offloading and data transferring, we found no study focusing and customising the algorithm for biomedical signals. In order to answer the second question, we proposed architecture of MCC-based telerehabilitation platform to support various types of sensors. The analysis oriented decision support system (AODSS) and the security service layer (SSL) are also integrated in the platform. The former utilises the huge amount of data to analyse the performance of patients from different levels of granularities, while the latter protects the information of data spanning from acquisition, storing, and transferring throughout the platform. These two are research challenges in conventional clinical decision support system (CDSS) [14] and will be discussed in Sections 4.5 and 4.6, respectively.

The rest of the paper is organized as follow. Section 2 lists related work, followed by our main contribution, the encoding scheme, and approach to optimise the energy of computation and data handling, in Section 3. Further more, an overview of the system architecture, as well as the concepts of AODSS and SSL, is presented in Section 4. Additionally, the simulation and real-data experiments regarding the multilevel encoding scheme are presented in Section 5 where the demonstration of the interfaces of the telerehabilitation platform is included. The final section presents a discussion on the overall approach and concluding remarks.

2. Related Work

A number of studies [2] have been conducted in the past few decades on developing efficient telerehabilitation systems. Initially, most of the telerehabilitation was based on the plain old telephone systems, also known as POTS [15]. As the technology advanced, more sophisticated methods were introduced into the telerehabilitation process. For instance, in 2002, the utilization of interactive video was proposed by Clark et al. [16] in order to provide teletherapy specifically for stroke patients where two videophones were set-up both in the therapist's office and at the patient's home, enabling the communication between both parties for that particular physical rehabilitation. Using this method, the patient was able to perform the walking activities as well as take care of themselves at home independently. In another reported work by Liu and Miyazaki [17], a video-conference device based rehabilitation system was deployed between the staff in the university of Albert and the clinicians/students in rural areas. The deployed system was aimed at being utilized in the education field supervising the students and the clinicians rather than in the application of treatments where 96% of telehealth sessions were successfully reported. Russell et al. [18] have developed a telerehabilitation software based application to record the video from the video conferencing link in order to remotely monitor the patients with gait conditions. They have evaluated the proposed system in terms of the accuracy and the reliability where the outcome of the gait assessments results through the internet was very close to the traditional method with less than 1 point on Gait Assessment Rating Scale (GARS). More examples can be found in [19].

Due to recent developments in cloud and mobile cloud computing technologies, more applications have been developed in the telerehabilitation and e-Health arena. Popescu et al. [20] has integrated a PC, a Polhemus tracker, and a haptic control interface together with force feedback in order to create a platform for the telerehabilitation service between the patients and the therapist through remote monitoring. Other than that, a web based telerehabilitation system was developed by Reinkensmeyer et al. [21] to enable the poststroke patients to perform rehabilitation exercises while being monitored by the therapist through the therapy games activity where a force-feedback joystick was used to provide resistance. Additionally, a virtual environment with a motion capture device (approximately $8500) was designed by Holden et al. [22] to offer a telerehabilitation environment for patients at home and, at the same time, the therapists were able to monitor patients' performance and hold video-conference remotely. The improvement of assessment scores during clinical trials illustrated the effectiveness of this system. Moreover, Sultan [23] discussed about opportunities and challenges in cloud-based healthcare provisioning and gave an example of a successful model: the pilot e-Health system at Chelsea and Westminster hospital. Wu et al. [24] proposed two approaches to perform offloading on multiple servers for mobile healthcare systems. The first approach is optimized for stabilization and the second approach is optimized for energy efficiency. Liu and Park [25] specified

characteristics of e-Health applications and proposed an adaptive architecture design for e-Health Cloud, Computing and Networking Solutions. Hendrick et al. [26] proposed a platform that is able to support medical records of an entire country. He et al. [27] proposed a private cloud platform architecture which includes six layers utilizing technologies such as MQ, load balance, plug-in algorithm, and cloud-storage. Wu and Khoury [28] designed a web service broker that can facilitate the data exchange between healthcare providers in order to achieve a complete patient medical records prior to treatment.

More relevant to our contribution, a number of approaches have been raised to estimate the energy consumption for mobile devices in mobile cloud computing applications. Wolski et al. [29] introduced a framework for offloading decision making by taking the bandwidth between the remote and local computer into account. Furthermore, Karthik [30] proposed a method to estimate the energy consumption by considering the computational power and bandwidth. More examples can be seen in [31].

3. Multilevel Data Encoding Scheme

This section is dedicated for solving one of the limitations, namely, limited power for computing and data transferring, in applying mobile cloud computing in telerehabilitation field by utilising the characteristics of biomedical data collected from various types of sensors.

3.1. Protocol. Due to the limited battery resources in a mobile device, how to reduce the power consumption is an open question in MCC field. In this paper, we propose a multilevel data encoding scheme so that the data can be encoded differently to reduce the quantity and time during transfer, thereby reducing the power utilized to transfer data. However, different encoding schemes require various computational times. Generally speaking, if the amount of data after encoding is smaller, the time utilised to encode the original data is longer. Therefore, it is crucial to find an approach to determine which level of the encoding scheme should be utilised.

Figure 1 shows the data encoding schemes for two types of telerehabilitation exercises, including motion rehabilitation (left side of the triangle) and respiratory rehabilitation (right side of the triangle). The encoding scheme in the higher position indicates that the data amount after encoding is smaller than approaches in lower levels. For instance, motion trajectories involve three-dimension data in a period of time and elbow points encoding scheme reduces the number of samples in motion trajectories, thereby reducing the amount of data. Furthermore, shape model reduces the dimension from three to two and performance measurement enables a telerehabilitation exercise to be represented by a single number that is meaningful to clinicians. Similar concept can be applied to respiratory data captured from Doppler radar, where the raw data (two-dimension) for respiratory exercises is I/Q radar signal. After applying arctangent demodulation, the exercises can be encoded by a single dimension data.

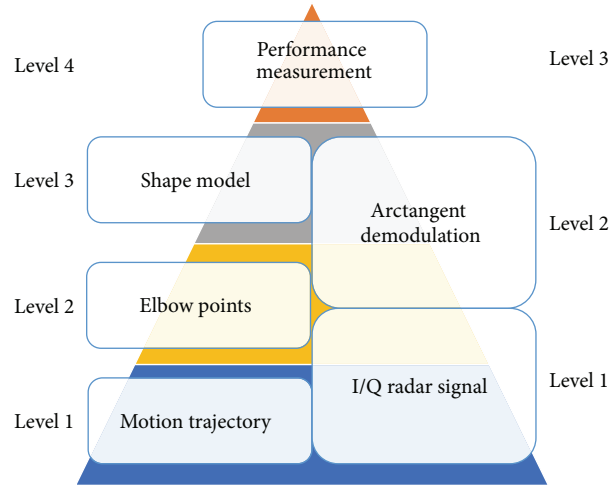


Figure 1: Multilevel exercise data encoding scheme.

Eventually, a performance measurement score can also be derived.

For instance, in human motion capturing, the details of each encoding scheme are introduced as follows.

(i) 3D Motion Trajectory. In the majority of the optical-based portable motion capture devices such as Kinect and Creative Senz3D, human motions are captured in terms of the positions of the joints in the forms of $\Gamma_n(t) = [x_n(t), y_n(t), z_n(t)]^\top$, where $x_n(t)$, $y_n(t)$, and $z_n(t)$ are 3D positions of nth joint ($n = 1, 2, \ldots, N$) on the x, y, and z axes at time t in the traditional Cartesian coordinate system. Compared to VGA video, a frame of 3D trajectories for N joints is $24 \times N$ bytes, where, for example, $N = 20$ in Kinect version 1 and $N = 16$ in Creative Senz3D. Since both videos and trajectories are collected from motion devices, we assume that the same power or energy is required to retrieve the data.

(ii) Elbow Point Technique. In a motion trajectory, each point has its own importance and its contribution to the shape of trajectory is different from others. An array of points lying on a straight line can be represented by only two points at the two end points of the line. Therefore, points at the middle of the straight line can be removed to reduce the computational cost and data storage. Points lying on a curve are called "elbow points" which are essential points to form the shape of the trajectory [32]. Discrimination of "straight points" and "elbow points" can be based on curvature. Curvature at a point $\Gamma_n(t)$ is defined as

$$\kappa_n(t) = \frac{\left\|\mathbf{v}_\mathbf{n}(t) \times \mathbf{a}_\mathbf{n}(t)\right\|}{v_n^3(t)}, \quad (1)$$

where

$$v_n(t) = \left[\left(\frac{\partial x_n(t)}{\partial t}\right)^2 + \left(\frac{\partial y_n(t)}{\partial t}\right)^2 + \left(\frac{\partial z_n(t)}{\partial t}\right)^2\right]^{1/2}. \quad (2)$$

A point is marked as an "elbow point" when its curvature is larger than a specific threshold ε. Conversely, a point is

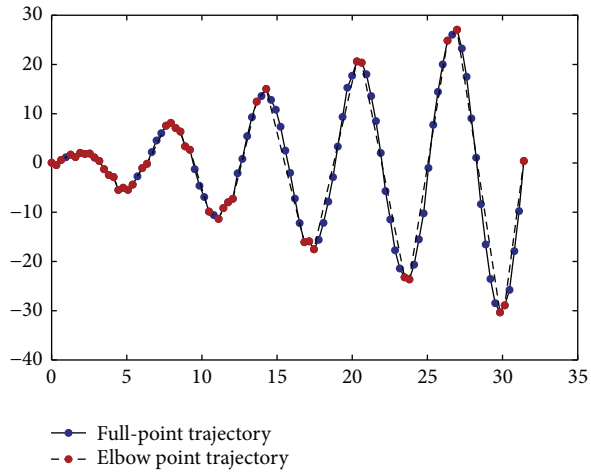


FIGURE 2: An illustration of the elbow point concept. Red points are elbow points when $\kappa > 0.05$. Blue points with curvature less than 0.05 are removed from the trajectory.

marked as "straight point" if its curvature is larger than or equal to zero and less than the specific threshold ε. In elbow method, points with curvature $\kappa < \varepsilon$ will be removed from the trajectory. The original trajectory can be approximately reconstructed from the new trajectory if the coordinates of the elbow points and their sequential orders are determined. The new trajectory which includes only elbow points obviously requires less computational cost than the original trajectory. We illustrate this technique in Figure 2 for the 2D case. The technique can be applied in 3D case [32]. Here, we plot the function $f(x) = x\sin x$ where $x_1 = 0, \ldots, x_{100} = 10\pi$ and $x_{n+1} - x_n = x_n - x_{n-1}$. As the figure shows, the new trajectory formed by elbow points is almost identical to the original trajectory with all the points. In this example, we use the value of 0.05 for the threshold ε, and hence half of the points from the trajectory have been removed. Depending on the application, this threshold value can be chosen accordingly.

(iii) Shape Model. To tackle some situations in which the bandwidth is insufficient to transfer elbow points, shape models can be further derived to encode motion trajectories. Apart from the curvature in (1), torsion is needed since the motion trajectories are in three dimensions. It is derived as

$$\tau_n(t) = \frac{\left(\mathbf{v_n}(t) \times \mathbf{a_n}(t)\right) \cdot \mathbf{j_n}(t)}{\left\|\mathbf{v_n}(t) \times \mathbf{a_n}(t)\right\|^2}, \tag{3}$$

where

$$\mathbf{j_n}(t) = \sqrt{\left(\frac{\partial^3 x_n(t)}{\partial t^3}\right)^2 + \left(\frac{\partial^3 y_n(t)}{\partial t^3}\right)^2 + \left(\frac{\partial^3 z_n(t)}{\partial t^3}\right)^2} \tag{4}$$

is the jerk of the motion trajectory.

From the derivation, it is observed that the motion trajectory is encoded from three dimensions (X, Y, and Z) to two dimensions (κ and τ).

(iv) Performance Measurement. Two methods can be utilised to encode and measure the performance of the telerehabilitation exercises, including smoothness based and elbow point based measurements. As for the former [33], it is specially designed for telerehabilitation service users with neurological movement disorders, such as dyskinesia, which involve large amplitude involuntary movements, which leads to less smooth motion trajectories than healthy people. Due to the fact that the shape model [34] is very sensitive to noise in motion trajectories, the submovements and jerky movements are also shown in the shape model. As a result, the entropy of the shape models of these trajectories are computed to represent the severity of involuntary movements, thereby indicating the ability of patients to perform telerehabilitation exercises given by their therapists. Another method can be used for this encoding scheme which is the algorithm in [32]. In this approach, the authors used longest common subsequence (LCSS) to match the trajectory from the patient and the corresponding one from the therapist (model motion trajectory) and gave a score for performance measurement.

Taking remote monitoring of human respiratory function using Doppler radar is another example to demonstrate the multilevel data encoding scheme. The methods used to perform encoding in various levels are shown as follow.

(i) Raw Data of In-Phase and Quadrature-Phase Signal. From a direct-conversion architecture of continuous wave Doppler radar, the motion from the abdomen due to inhalation and exhalation activities during breathing causes a shifting in the transmitted wave in terms of phase. Thus, the reflected signal received at the receiving ends will be a phase modulated signal that consists of the information of the motion during breathing. The received signal is denoted as

$$R(t) \approx \cos\left(2\pi f t - \frac{4\pi d_0}{\lambda} - \frac{4\pi y(t)}{\lambda} + \phi\left(t - \frac{2d_0}{c}\right)\right), \tag{5}$$

where ϕ is the phase noise of the signal source, a nominal distance of d_0, and the time varying displacement of $y(t)$ due to corresponding respiration activities. To overcome the null problem, a quadrature receiver is used where the signal will be further represented with the I and Q signals denoted as

$$\begin{aligned} I_B(t) &= \cos\left(\theta + \frac{4\pi y(t)}{\lambda} + \Delta\phi(t)\right), \\ Q_B(t) &= \sin\left(\theta + \frac{4\pi y(t)}{\lambda} + \Delta\phi(t)\right). \end{aligned} \tag{6}$$

(ii) Arctangent Demodulation. Depending on the aim of the application, I and Q signals can be combined using arctangent demodulation for a better accuracy and system requirements (system with bandwidth constraint). The combination of both signals can be represented as

$$x(t) = \tan^{-1}\left(\frac{Q_B(t)}{I_B(t)}\right) = \frac{4\pi x(t)}{\lambda} + \Delta\phi. \tag{7}$$

Table 1: Notations used in the process of estimating power consumption for data encoding and transfer. Here $i = 1, 2, 3, 4$ for motion telerehabilitation and $i = 1, 2, 3$ for respiratory telerehabilitation.

Notation	Unit	Description
d	Byte	The size of memory occupied by a double value
D_i	—	Dimension of encoded data of each frame with ith encoding scheme
L_i	Frame	The length of exercise data encoded by the ith encoding scheme
N	—	The number of monitored points in the motion. $N = 1$ for respiratory monitoring
S_i	Byte	Size of encoded data at the scheme ith
u_i	Watt	The energy consumed by the mobile device to encode 1 frame at the scheme ith
v	Watt	The energy consumed by the mobile device to transfer encoded exercise data to the cloud
P_i^L	Watt	The energy consumed by the mobile device to encode entire trajectory at the scheme ith
P_i^T	Watt	The energy consumed by the mobile device to transfer entire encoded data to the cloud at the scheme ith
B	kbps	The speed of uploading data to the server

3.2. Determine Encoding Level. The introduction of various encoding approaches naturally raises the question of determining the encoding level that should be used depending on the computational power and the speed used to upload data to the cloud from the mobile device. The work in [30] introduces an approach for offloading decision by estimating how much power is used for computations and transfer of data. We adopted this method to determine which encoding level should be utilised with respect to the bandwidth. Notations utilised in the estimation process are shown in Table 1. Unlike the approach introduced in [30], the computation time in the server is not considered in our case. The reason is that the data communication between the server and the mobile device uses asynchronous channels which means the channels are not blocked when the server is engaged in the computation. Therefore, there is no idle time in the mobile device. Further in our case, it is hard to estimate the number of instructions required by the computation for data encoding. As a result, we directly record the numbers of unit energy consumption (u_i) utilized to perform encoding for each frame of data, which can be retrieved automatically by the system.

The formula to calculate data size is

$$S_i = d \times D_i \times L_i \times N. \tag{8}$$

The formula used to calculate local computational power with scheme i is

$$P_i^L = P_{i-1}^L + u_i \times L_{i-1} \times N, \tag{9}$$

where $u_1 = 0$; u_2 is the average power consumption to determine whether a point is an elbow point and it includes the calculation of curvature; u_3 is the average power consumption to calculate torsion value of 1 point; u_4 is the average power consumption to calculate the performance.

The formula used to calculate data transfer power is

$$P_i^T = \frac{S_i}{B} \times v, \tag{10}$$

where v is the energy consumption for uploading data in 1 second and B is the network speed. Table 2 summarises the power consumption of various encoding schemes and encoded data transfer for motion telerehabilitation.

The following cost function is used to determine which encoding scheme is to be used:

$$E(i) = \underset{i}{\operatorname{argmin}} \left(P_i^L + P_i^T \right). \tag{11}$$

In some cases, if $|E(i_1 - E(i_2)| < \epsilon$ and $i_1 < i_2$, we select $i = i_1$ so that more data can be transferred to the cloud by consuming similar power. Here ϵ is a small constant value.

4. System Architecture

The severity of certain medical conditions can be evaluated from various aspects. For example, respiratory disorders [35] (i.e., dysfunctional respiratory and chronic obstructive pulmonary disease) can be assessed through breathing and motion monitoring and, in a contrasting condition, both hand and limb are affected by stroke. Therefore, in the first example, we may use chest straps or a Doppler radar to detect the respiratory function with inertial measurement unit (IMU) (BioKin in our case [36]) to monitor limb movements. In the second example, Kinect and Creative Senz3D camera can be utilised to monitor the movement of limbs and hands for assessment purposes. These examples indicate that it is essential to have a telerehabilitation platform catering for various sensors aiming at capturing various physiological measures. In this section, we propose architecture for a MCC-based telerehabilitation platform, as well as the implementation details of the prototype system for testing purposes. Although we utilise Kinect, Creative Senz3D camera, Doppler radar, and BioKin as sensors in the prototype system, sensors such as EMG and ECG can also be supported with minimal modification.

4.1. System Platform. As for the platform to deploy the server side of the proposed system, we selected Microsoft Windows 2012R2 for its stability, ease to configure, and being well supported by various cloud computing providers, such as Amazon EC2. In addition, the programming language utilised to develop the overall system was C# due to the following three reasons:

Table 2: Power consumption of various encoding schemes and encoded data transfer for motion telerehabilitation.

	i	D	L	Data size	Energy consumption of local computation	Energy consumption of data transfer
Original trajectory	1	3	L_1	$24 \times L_1 \times N$	0	$24 \times L_1 \times N \times v/B$
Elbow points	2	3	$\approx 0.5 \times L_1$	$12 \times L_1 \times N$	$u_2 \times L_1 \times N$	$12 \times L_1 \times N \times v/B$
Shape model	3	2	$\approx 0.5 \times L_1$	$8 \times L_1 \times N$	$P_2 + u_3 \times 0.5 \times L_1 \times N$	$8 \times L_1 \times N \times v/B$
Overall performance	4	1	1	8	$P_3 + u_4 \times 1 \times N$	$8 \times v/B$

(i) The sensors utilised in our telerehabilitation system support C#, especially the Microsoft Kinect, which has SDK (v2.0) for C#.

(ii) C# is well supported by Windows operating systems.

(iii) The Windows Communication Foundation (WCF) can be integrated as the basic communication framework for the overall system, which can be easily developed by using C#.

Therefore, as per the selection of deploying platform and programming language, the integrated development environment (IDE) utilised to implement the whole system is Microsoft Visual Studio 2013. In order to use the latest functions of C#, we adopted .Net 4.5 as our common language run-time (CLR).

4.2. Web Service Style. To provide web services, two principles in [37] (or called tools to design web service in [38], architectural style [39], and so on) were extensively utilised, including Simple Object Access protocol (SOAP) [40] and Representational State Transfer (REST) [41]. Although both are able to be the architecture for data communication in web service, the latter is more suitable in our system for its capability of reducing the delay. According to [37], the message size and processing and transmission time of REST web service was 9 to 10 and 5 to 6 times, respectively, shorter than that of SOAP based ones, since REST protocol does not need to format payload data as formal as SOAP, thereby reducing the overhead and processing time to parse XML. Eventually, the delay of the whole system can be minimized from a data communication perspective.

Due to the selection of REST, which is HTTP-based, we selected WSDualHttpBinding as binding method for contracts in WCF for its ability to provide asynchronous communication by simulating duplex channels to separate the requests and responses. Therefore, even with some delay in the cloud, the channel is less likely to be blocked compared to single channel model.

4.3. System (Hardware) Overview. From Figure 3, we can see that there are three main components in the telerehabilitation platform: therapist side, server side, and patient side. Except for the server side, there is only one instance of the other two components in the figure, which can actually be multiple instances simultaneously, representing a number of therapists and patients accessing the telerehabilitation services at the same time.

At current stage, due to the limitation of SDKs and drivers of sensors, especially Kinect, Creative Senz3D, and Doppler radar, they are unable to connect to tablet and mobile phone. However, it is noteworthy that a trend has been seen, in recent years, to integrate these sensors into mobile devices. For example, Dell Venue 8 7000 series tablets have integrated Intel RealSens technology to support depth cameras which is conceptually similar to the concept of Kinect. HTC M7 smartphone integrated two cameras to provide depth images which also is a similar concept to the LeapMotion or similar in function to Kinect. Therefore, our proposed telerehabilitation platform not only caters to the existing sensors, but also can be utilised when the sensors are integrated into mobile devices.

Four types of sensors are shown in this paper as examples. Firstly, Microsoft Kinect version 2 is one of the sensors used in our telerehabilitation system. The main function of this device is tracking the 3D positions of 25 built-in joints or multiple self-defined markers with the sampling rate of 30 Hz. The details of the specification of the second version Kinect can be retrieved in [42]. Secondly, BioKin [43, 44] is aimed at capturing human movements with wearable IMU sensors to facilitate exercises typically for remote telerehabilitation. As an ambulatory system possibly used in nonclinical settings, BioKin can provide complementary services to communities with limited access to infrastructure such as gait laboratories. It consists of an inertial sensor, self-contained triaxial magnetometer providing 150 Hz sampling frequency. Thirdly, the Creative Senz3D is an optical sensor which is capable of acquiring depth images and specialised for human body tracking and gesture tracking. While Kinect is designed for full-body capture, the Creative Senz3D is optimised for short-range gesture interaction and, consequentially, it is perceived as a better alternative to hand measurements. Depth images of the Creative Senz3D have the resolution of 640 × 480 pixels and are refreshed with the frequency of 30 Hz. Lastly, a 2.4 GHz quadrature based Doppler radar is a noncontact measurement device that is capable in acquiring the human physiological sign such as respiration and heartbeat motion. Measurements of the signals can then be derived into different forms of information for further analysis. System measurement consists of the time series of in-phase (I) and the quadrature-phase (Q) signals which is then sampled at 1000 Hz. The sampling rate of the system can be reduced to as low as 30 Hz depending on the aims of the application. The obtained measurements can be further used for but not limited to information extraction [35, 45] and refining

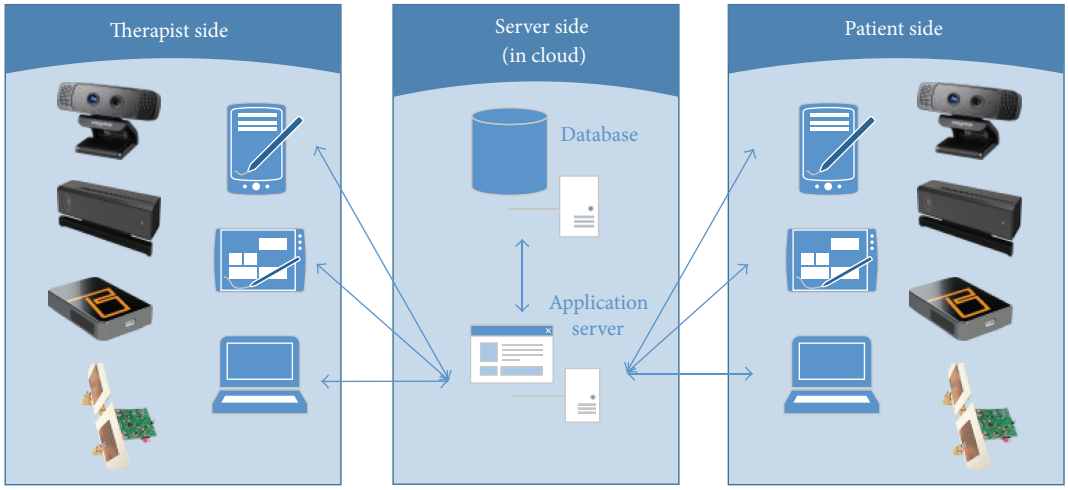


Figure 3: The overview of the telerehabilitation system.

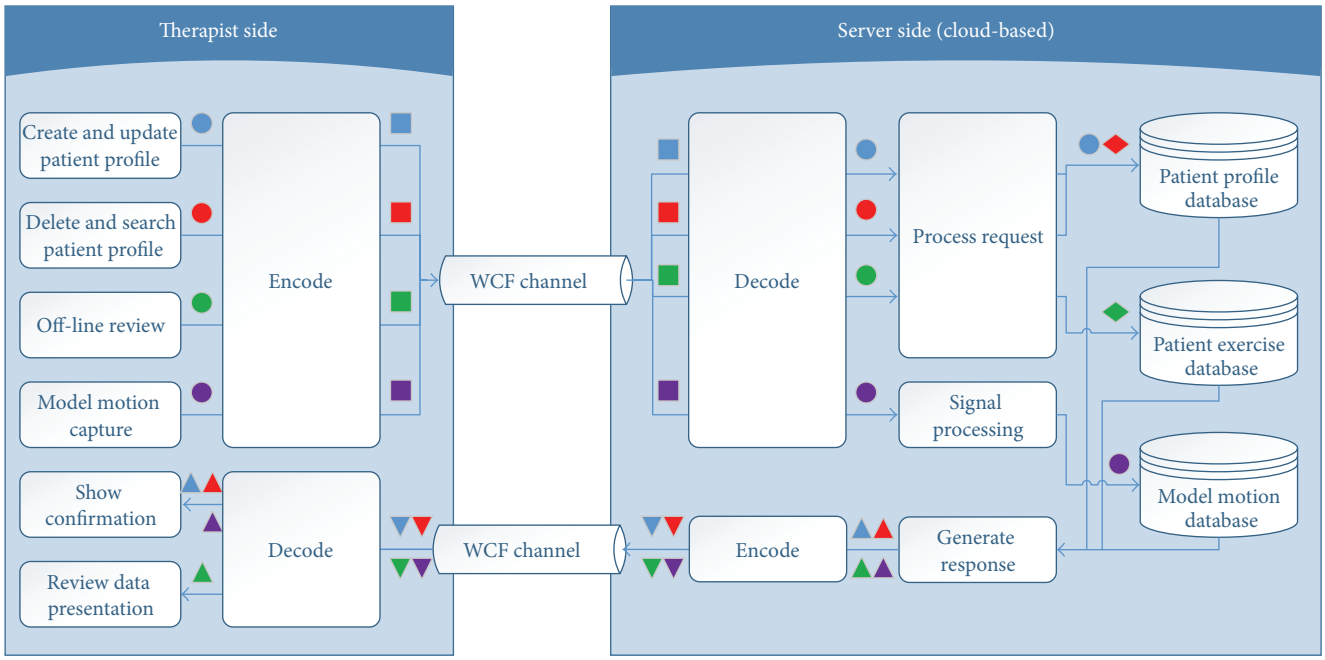


Figure 4: Data flow model in therapist side (except for online review).

diagnosis procedure and as external input to trigger the biofeedback mechanism such as in sleep apnoea and sudden infant death syndrome (SIDS) detection applications.

These sensors can be used by the therapists to record model motions and respiratory data as examples or references for patients to follow and compare, while collecting data for exercise performance assessment and further analysis in patient side.

4.4. System (Software) Architecture Overview. In this subsection, a detailed software architecture of the system is provided as follows.

(i) Therapist Side. Primary focus of this side is on the inclusion of patient profile management, building exercise models (example can be found in Figure 10), and visual (online or off-line) review of the exercise data and analysis result collected from various sensors and AODSS (refer to Section 4.5). Exercise models built in this side have two major aims. First of all, the models can be downloaded by patients and utilised as a guidance in performing telerehabilitation exercises. Secondly, models built by therapists can be used as references for telerehabilitation exercise performance evaluation. As for the data flow, except for life streaming, it is introduced in the patient side.

Figure 4 shows the data flow between the cloud (server side) and the therapist side. Symbols in the graph with different colours indicate the data and requests from various sources and their corresponding responses. Here squares and downward triangles are the encoded version of data

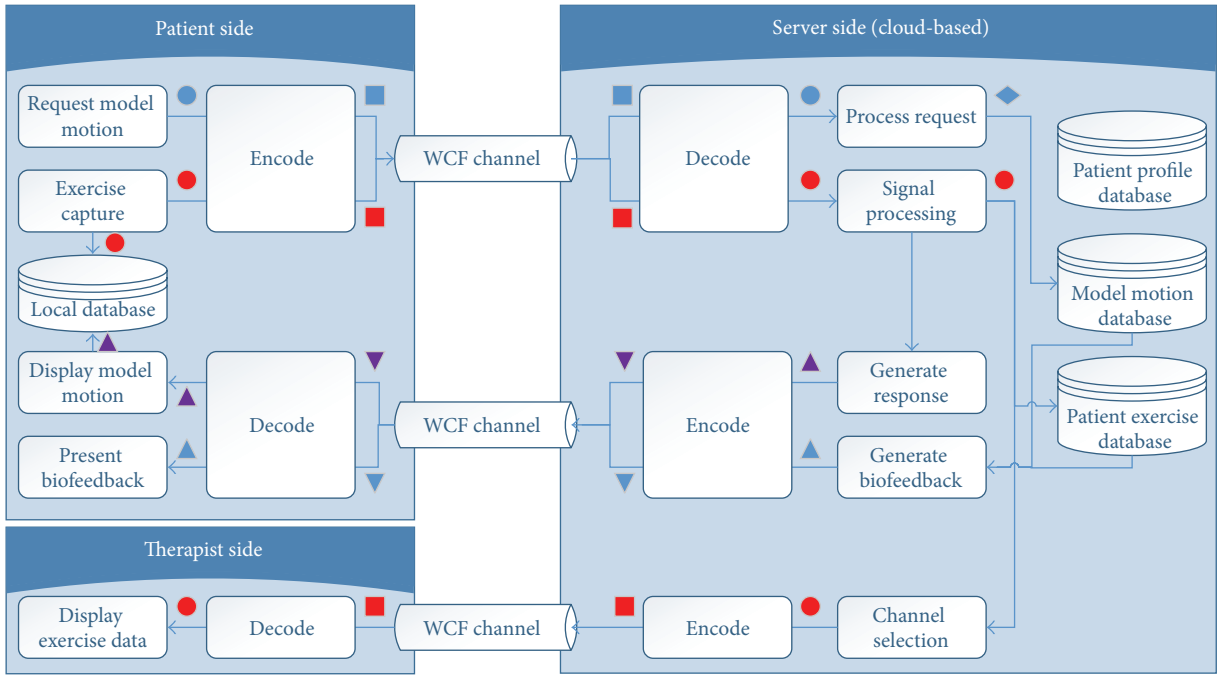


Figure 5: Data flow model in patient side (with online review for therapists).

represented by circles, while upward triangles and diamonds represent database operations without storing new data. Additionally, as for the responses, there are two methods to display information. One is showing a confirmation statement to notify the therapists whether the requested operation is successful or not. The second approach is visually displaying information obtained from the cloud, such as histograms. In addition, the data flow of off-line review (with green symbols) in Figure 4 is an abstract process and detailed information is discussed in Section 4.5.

(ii) Patient Side. The patient side of our platform provides interactive telerehabilitation services and passive exercise monitoring capabilities. The data flow is shown in Figure 5. For interactive telerehabilitation, first of all, the patient requires the exercise model created by a therapist from the cloud (shown with blue symbols), which is later utilised in the rehabilitation exercises as references and streamed to the patient's mobile devices (with purple symbols). After that, the performed exercise motions of the patient are recorded and sent to the cloud (with red symbols). It is noteworthy that, instead of sending video or audio data like typical telerehabilitation systems, the encoded data is sent in our system.

When the cloud (server side) receives the recoded exercise information, signal processing techniques are applied to filter out the noise and also to extract relevant features. The data is stored in patient exercise database for off-line review and also used to provide corresponding biofeedback particularly for performance measurement and assessment through the comparison between patients' acts and therapists' models. The biofeedback is denoted with blue triangles and presented in various forms (refer to Section 4.7). Lastly, if therapists have enough bandwidth and choose to review patients exercise online, they just have to register a channel, through which the processed motion information is forwarded by the cloud to the therapists. As for the passive exercise monitoring, the data follow the path indicated by red symbols.

In addition, to solve the disconnection issue in mobile terminals, a small local database is maintained in the patient side to store the most recently used model motion information and patients' exercise motion data. Therefore, even when patients are unable to connect the Internet temporarily, they are still able to use the telerehabilitation system. When the connection is established again, patients' motion data can be synchronised to the main database in the server side for further analysis. Indeed when the Internet is not available, patients are unable to receive biofeedback and performance measurement updates if the computational power in their mobile device is insufficient.

4.5. Analysis Oriented Decision Support System. In this subsection, we discuss the concept of analysis oriented decision support system (AODSS), which is a combination of the concept of the serviceoriented decision support system (SODSS) [46] and the clinic decision support system (CDSS) [14]. As the AODSS integrated in the MCC-based telerehabilitation platform, it contributes to the establishment of mobile CDSS mentioned in [14], which was considered a research challenge in CDSS. The contribution of the paper to AODSS is that data is analysed with different granularities, which can be collected from various types of sensors and stored in various databases in the cloud. This method is embedded into the data flow (with green symbols) shown in Figure 4.

The concept of AODSS is illustrated in Figure 6. The first column, including various wearable or nonwearable sensors, shows the source of data, which is not the process of data

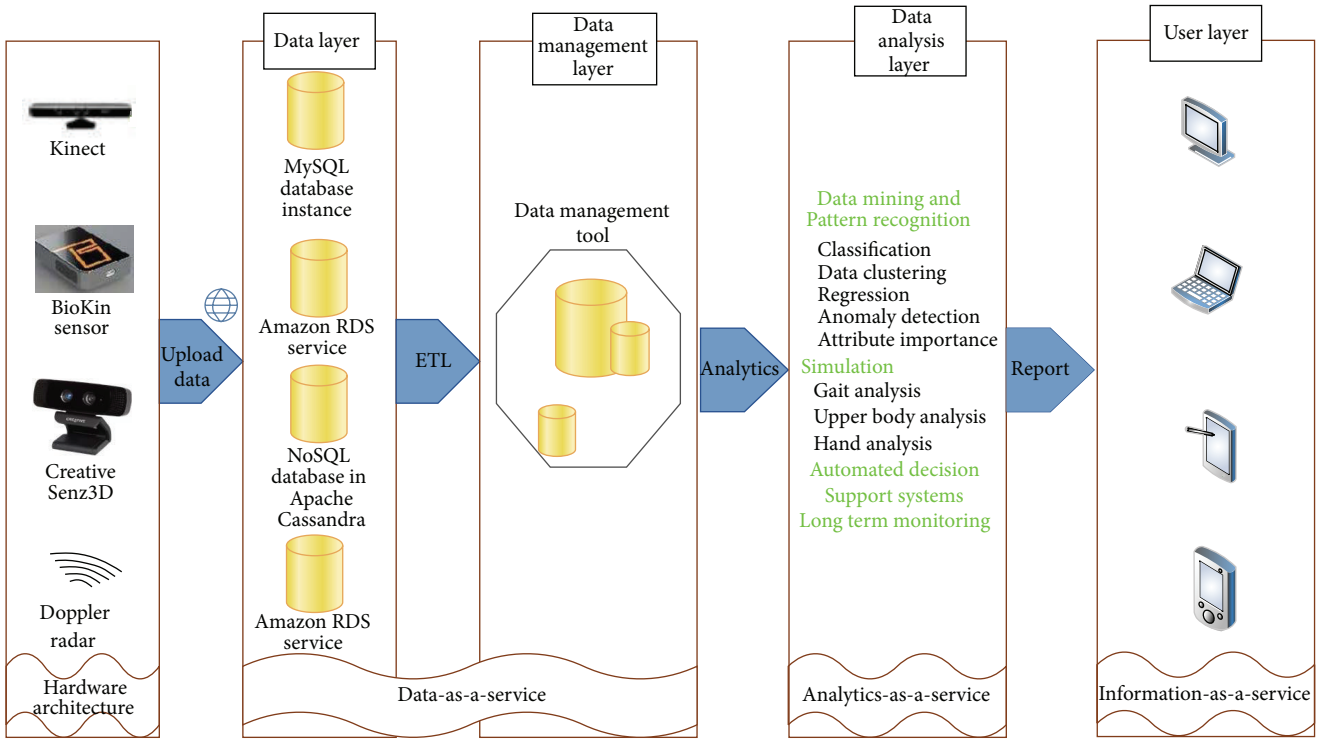


Figure 6: The conceptual architecture for analysis oriented decision support system.

flow of information query and response between therapist side and the sever side. Data from these sensors are stored in different databases in the cloud which is highly likely to be distributed into different servers to maintain the response speed with a large number of queries. After raising a query from the therapist side (user layer) to the data management layer, the data extraction tool retrieves related data from some databases (in data layer). Then the data is further processed by using various integrated data mining tools (in data analysis layer) and mining algorithms, such as clustering, classification, regression, attribute importance, anomaly detection, association, and feature extraction. Eventually, the data is visually presented to the therapists to assist them to make further decisions.

The key point in this AODSS is that data stored in database is encoded with various encoding scheme. As mentioned above, the selection of the encoding scheme is determined by the computational power of the mobile device and the speed of the Internet. Our AODSS is able to not only analyse data from various types of sensors but also provide supportive information in various granularities to therapists based on the encoding levels of data sources.

Use stroke telerehabilitation as an example. Since the kinematic performance of a poststroke patient can be assessed through the movement of limbs and hands, AODSS first retrieves related data from two databases, including Kinect and Creative Senz3D databases. Due to the fact that data used by AODSS is encoded with various encoding schemes, AODSS is able to provide results in various granularities. For instance, AODSS can provide a histogram showing the change of performance (computed by the performance measurement algorithms mentioned in Section 3.1) of a patient in telerehabilitation sessions during a day, a month, or even longer period. After receiving the analysis results, the therapist will have a general idea about the effectiveness of the therapy and the progress of the patient. If the therapist wants to know the movement patterns of the patient in each session, AODSS is also able to analyse the encoded data at motion trajectory, elbow points, or shape model level and generate a report to show the details in each session. These information assists the therapist to understand the detailed reason why the patient is underprogressing or improving. Thus, the therapists can reevaluate the exercise components in the session to suit the patient's capabilities to achieve better rehabilitative outcomes or have the confidence to encourage the patient to stick with the therapy. Apart from that, AODSS fuses data from both Kinect and Creative Senz3D to show the performance of the poststroke patient in both limb movements and hand movements. Therefore, the therapist can receive a more comprehensive information and produce better relevant decision for the therapy session.

4.6. Security Service Layer. As mentioned previously, security and privacy are one of the challenges faced in CDSS. Therefore, AODSS is inevitably impacted by this challenge. Security service layer (SSL) introduced in this subsection is a concept that has the potential to solve this challenge in the proposed AODSS, as well as the MCC-based telerehabilitation platform.

Since many parties such as patients, clinicians, healthcare administrators, insurers, and researchers are involved, an end-to-end security control will be applied to enhance

the protection of data. The security association is managed by a Security Association Manager to coordinate the communication groups [25]. The Security Association Manager includes Security Policy Collections, Security Association Flows, and Distributed Logs. The Security Policies Collection specifies which security level and which rules need to be used in associations with the types of parties. The Security Association Flows contain routing algorithms, keys, encryption schemes, protocol modes, and flow-level lifetime. The Distributed Logs store the logs of both end points in order to avoid fragment of logs stored in different repository. The Security Association Manager can collaborate with external Certificate Authority [47] to enhance authorization processes. This cooperation allows a patient or the service provider to rely on their identity provider to provide other e-Healthcare providers with only the specific data necessary to complete the transaction.

4.7. Optimize Biofeedback. Traditionally, the feedback with regard to the rehabilitation exercises is given directly from therapists, which is the most effective approach. Therefore, this method is implemented in our system as a biofeedback option. However, it is only available when the therapist reviews the exercises of the patient online so that the auditory message is routed by the server side from the therapist to the selected patient and vice versa for communication. It means that this method relies heavily on the large bandwidth, which is not always available for MCC. Therefore, in our system, we have three other types of biofeedback, which can be selected according to the availability and the bandwidth of the allocated network connection.

First of all, since the model motions recorded by therapists were utilised to guide patients to perform telerehabilitation exercises, patients are able to receive the visual biofeedback directly by looking at the differences between their motion trajectories and those from the model motions. However, due to the fact that the motion trajectories on the screen are always in 2 dimensions and the third one is unable to be observed by patients, the top down view of both the models and patients' motions are presented in the other indow to illustrate the differences in the third dimension. By looking at the screen, patients are able to see the gap between their motions and the models, thereby correcting their motions in time. Since the model motion is always available, either from the cloud (when the speed of the Internet is fast enough) or from the local temporary database (when the bandwidth is small or the Internet is disconnected), this type of biofeedback is always available.

Secondly, auditory biofeedback is also an option in the introduced platform to correct patient's telerehabilitation exercises. For instance, with the respiratory exercises, music with various rhythms is given so that the patient is able to modify the frequency of the breathing by following the rhythm of the music. The same concept is utilised for motion rehabilitation exercises where fast rhythm indicates that the patient should move the limbs or other body parts involved in the telerehabilitation session faster and the slow rhythm tends to slow down the patient's movement. To modify the rhythm of the music, the speed of the patient's movement is derived from their motion trajectories in the server side and is compared with that of the model motion. Eventually, a ratio is generated and sent to the patient side to change the rhythm.

Lastly and most importantly, although we utilise performance measurement as an encoding scheme (refer to Section 3), this value can also be utilised as a feedback indicating the performance of doing specific rehabilitation exercises. As is mentioned in the previous subsection, two approaches (including elbow point based and entropy based) could be utilised to represent the performance of a rehabilitation exercise session. This feedback not only gives both the patient and the therapist an overview about the ability of the patient to perform certain tasks, but also is able to stimulate the patient to perform exercises more frequently, thereby achieving higher performance measurement.

5. Numerical Results and Platform Demonstration

5.1. Computer Simulation for Multilevel Encoding Scheme. In this subsection, a numerical example is presented to demonstrate the effectiveness of the multilevel encoding scheme for motion exercise monitoring in saving energy in mobile devices. First of all, we generated 18 motion trajectories with $L_1 = 1000$ for each exercise. We further assumed $u_1 = 0\,\mu$J, $u_2 = 8.8\,\mu$J, $u_3 = 21\,\mu$J, and $u_4 = 0.6$ mJ. These assumptions were based on execution time for each specific task which directly related to the complexity of the computations and transmission. We also assumed that we use a connection setting which cost $v = 50$ mJ for transferring data in 1 second. We examined the total energy consumption for both local computations and data transfer for different uploading speeds from 0 kbps to 2000 kbps. The numerical experimental result is shown in Figure 7. In the figure, the minimum energy for the uploading speed from 0 kbps to 50 kbps can be achieved if we use level 4 of encoding scheme. Similarly, we use level 3 for uploading speed from 50 kbps to 200 kbps, level 2 for uploading speed from 200 kbps to 1200 kbps, and level 1 for above 1400 kbps. At the uploading speed of 50 kbps, the total of energy spending for levels is 21.6 J, 10.9 J, 7.5 J, and 1.3 J. The difference of energy between levels can be up to 20.3 J and this means the use of encoding schemes can save up to 20 times of energy for low connection speeds. Indeed, for better connections, for example, uploading speed above 1400 kbps, sending all data to the cloud for processing is the best method in terms of saving energy and preserving information.

5.2. Real-Data Experiment for Multilevel Encoding Scheme. To further illustrate the performance of the proposed multilevel encoding scheme, we performed a preliminary real-data experiment with simulated motion trajectory data and four various encoding methods. In this experiment, software named BatteryMon (V2.1 build 1008) and NetBalancer were utilised to monitor the energy consumption and to control the speed of the Internet connection. Furthermore, we implemented the encoding algorithms and data transferring program in C#. The experiment was done on a laptop with CPU of Inter Core i7-3740QM and Wifi card of Intel Centrino Ultimate-N 6300AGN. To eliminate the influence of other

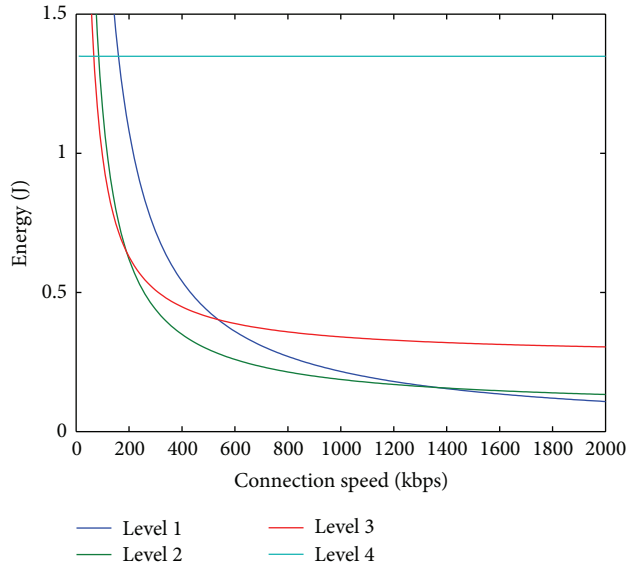


Figure 7: Numerical experimental result.

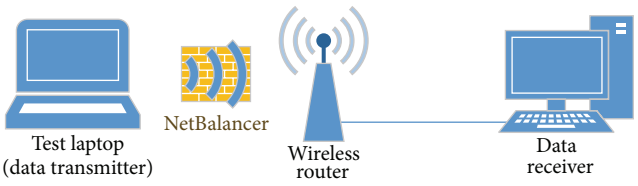


Figure 8: Setup of the real-data experiment.

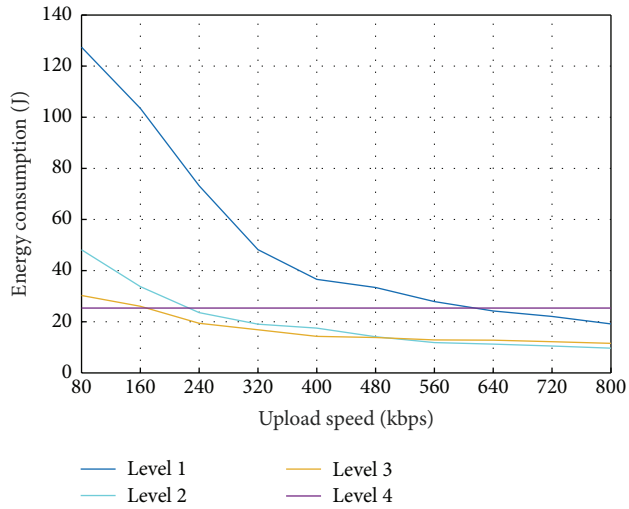


Figure 9: Result of real-data experiment for the multilevel encoding scheme.

software, first of all, BatteryMon was initialised for half an hour without running any unnecessary program to estimate the energy consumption of the laptop in idle state. All the following measurements were subtracted by this energy to compute the power utilised in order to compute the proposed encoding methods or to transfer the data to the cloud.

After that, a 3D trajectory with length of 1000000 frames was collected and further encoded with the other three encoders. Each of the encoding was repeated for 10 times (reliability test) where the average energy consumption of computing for each encoder was recorded in terms of per frame.

Furthermore, the setup of the real-data experiment is shown in Figure 8. We deployed the data receiver component of the data transferring program in a desktop connected to a wireless router with a network cable to secure the speed of the data transfer. Additionally, the laptop is connected to the router with the Wifi card like a normal mobile device. Moreover, NetBalancer was used to control the Internet upload speed of the laptop to simulate the environment with different conditions of the Internet. We limited the upload speed from 80 to 800 Kbps with a step of 80 Kbps for testing because the upload speed on 3G/4G is about 0.45 Mbps to 1.93 Mbps [48]. Eventually, we recorded the average energy utilized to transfer one frame of the encoded data with each respective speed.

Lastly, we computed the total power consumption for each encoding approach and various uploading speeds for a trajectory with 10000 frames. The result is shown in Figure 9.

From the result, a similar trend as the simulation was observed. When the upload speed was smaller than about 160 Kbps, level 4 encoder was the best option since it consumed the lowest energy to encode the motion data and transfer the result to the cloud, while the level 3 encoding approach should be selected when the upload speed of the Internet was ranging from 160 to 480 Kbps and, subsequently, level 2 encoder should be utilised. Using real experiment's data, it shows that level 1 encoding method always consumes more energy than level 2 and level 3 encoder in this scenario.

5.3. Telerehabilitation Platform Demonstration. This subsection shows snapshots of the implemented telerehabilitation platform. First of all, the example of the model making function in therapist side is shown in Figure 10.

In this figure, it can be seen that a therapist is able to make a specific model for a particular patient according to the condition of this patient. One or multiple joints can be selected as the important joints in this exercise model. Lastly, the functions of replay and crop of the recorded therapist's model motion were incorporated in the system to adapt the needs of the therapist.

The example of the patient side is shown in Figure 11. There are five components in this figure. The left side of the figure shows the front view of the patient skeleton with the guidance trajectory. The top right of the figure shows the top view of the tracking joint (yellow circle) and the guidance point (light blue circle). The middle right chart shows the history of the performance measurement for a period of time and the meter in the bottom right shows the performance's level of the current session. Additionally, the message where the *Start* word appears shows the incoming instructions as a form of visual feedback. The aim of the exercise is to match

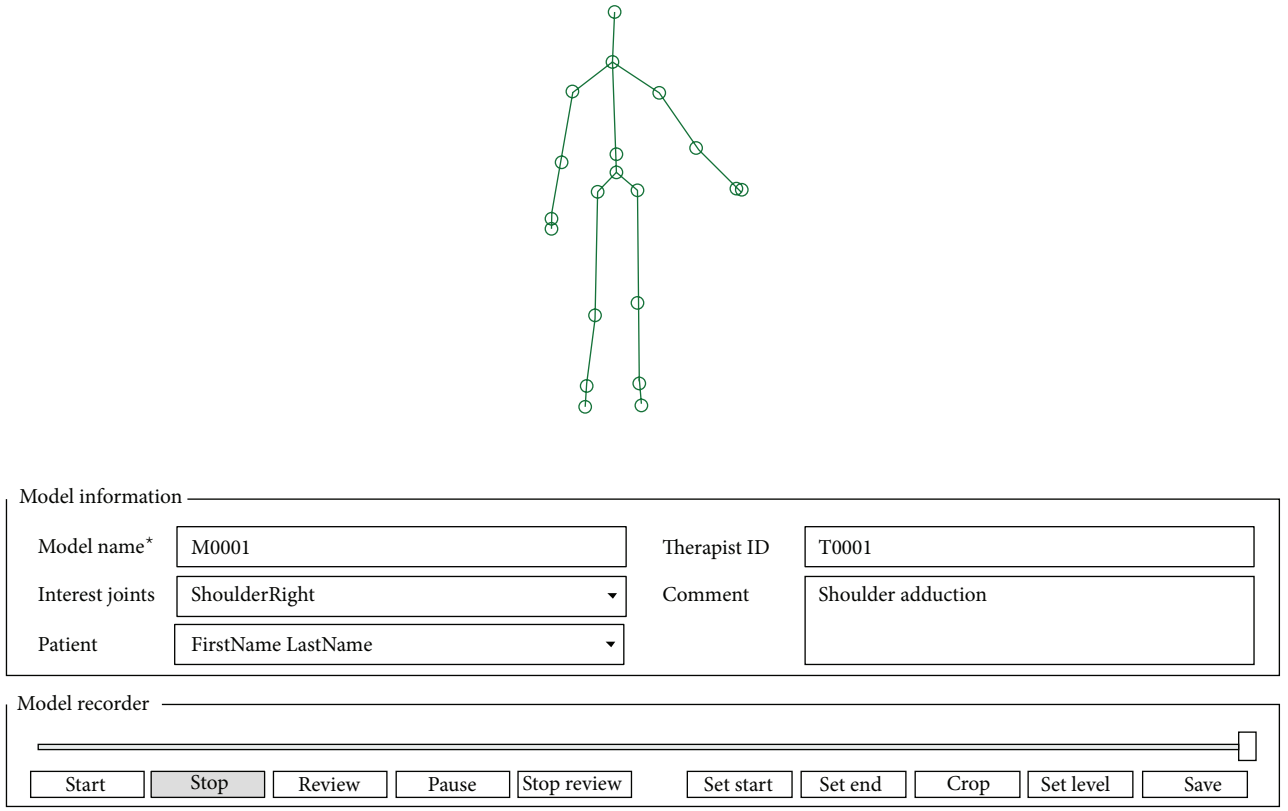


Figure 10: Example of making model motions of the therapist side with Kinect.

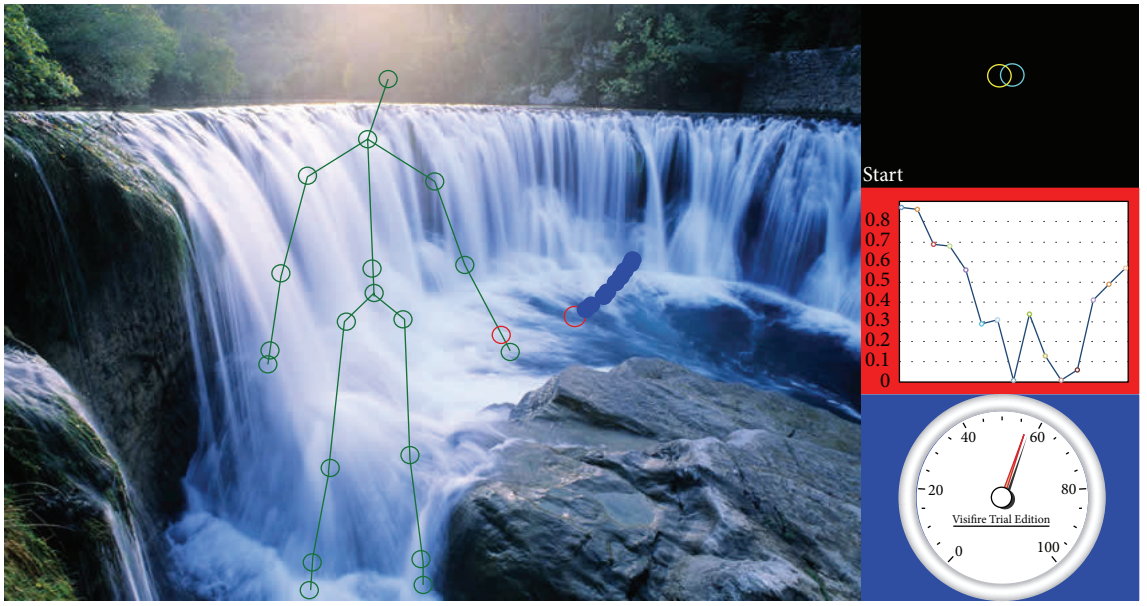


Figure 11: Example of the patient side.

the red circle on the skeleton onto the red circle on the guidance (blue trajectory).

If the Internet connection is available, the motion trajectories of the patient are encoded with one of the approaches introduced in Section 3 depending on the computational power of the mobile device as well as the upload speed of the Internet. After each session, the performance measurement, which is also the level 4 encoder, is computed either on

the local device or in the cloud platform. Eventually, this measurement is shown in the patient side as an overall biofeedback.

6. Conclusion

Optimising the limited battery resources on mobile devices to provide maximum time for patients to access the telerehabilitation services is a major concern in the implementation of a MCC-based telerehabilitation system. This paper introduced a novel multilevel encoding scheme to minimise the energy consumption in the mobile devices by taking the advantage of characteristics of biomedical signals. The computer simulations and experimental data have primarily shown the advantage of the proposed approach in selecting the suitable encoding approach for saving energy.

Apart from that main contribution, the concept of analysis oriented decision support system (AODSS) and a service security layer (SSL) were also introduced as an important component of the proposed telerehabilitation platform. Particularly, the AODSS provides therapists with supportive suggestions by analysing a huge amount of data collected from different types of sensors in various levels of granularities. As a result, more comprehensive results can be supplied to assist therapists to make further decisions. Furthermore, SSL secures the data in collecting, storing, and transferring within the platform.

Different sensor types were successfully tested separately in lab environment with healthy subjects to evaluate the feasibility of overall platform. Future studies will be focused on capture and analysis of clinical data with a larger patient group related to specific medical conditions, such as Parkinson's disease, stroke, and sleep apnoea, and extending the AODSS database to evaluate the support system. Moreover, various types of new sensors will be added into this system gradually in future clinical trials.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Authors' Contribution

Saiyi Li and Hai Trieu Pham made equal contributions to this paper.

Acknowledgment

This work is supported by the National Information and Communication Technologies Australia (NICTA) and Deakin University.

References

[1] A. Regalado, "Who coined 'cloud computing'?" *MIT Technology Review*, 2011.

[2] M. Rogante, M. Grigioni, D. Cordella, and C. Giacomozzi, "Ten years of telerehabilitation: a literature overview of technologies and clinical applications," *NeuroRehabilitation*, vol. 27, no. 4, pp. 287–304, 2010.

[3] D. Kairy, P. Lehoux, C. Vincent, and M. Visintin, "A systematic review of clinical outcomes, clinical process, healthcare utilization and costs associated with telerehabilitation," *Disability and Rehabilitation*, vol. 31, no. 6, pp. 427–447, 2009.

[4] M. Zampolini, E. Todeschini, M. B. Guitart et al., "Telerehabilitation: present and future," *Annali dell'Istituto Superiore di Sanita*, vol. 44, no. 2, pp. 125–134, 2008.

[5] C. R. Carignan and H. I. Krebs, "Telerehabilitation robotics: bright lights, big future?" *Journal of Rehabilitation Research and Development*, vol. 43, no. 5, pp. 695–710, 2006.

[6] T.-V. How, "Co-design of cognitive telerehabilitation technologies," in *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices & Services (MobileHCI '14)*, pp. 407–408, 2014.

[7] J. Maitland, M. McGee-Lennon, and M. Mulvenna, "Pervasive healthcare: from orange alerts to mindcare," *ACM SIGHIT Record*, vol. 1, no. 1, pp. 38–40, 2011.

[8] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[9] Amazon EC2, http://aws.amazon.com/ec2/.

[10] O. M. Giggins, U. M. Persson, and B. Caulfield, "Biofeedback in rehabilitation," *Journal of NeuroEngineering and Rehabilitation*, vol. 10, article 60, 2013.

[11] C. E. Koh, C. J. Young, J. M. Young, and M. J. Solomon, "Systematic review of randomized controlled trials of the effectiveness of biofeedback for pelvic floor dysfunction," *British Journal of Surgery*, vol. 95, no. 9, pp. 1079–1087, 2008.

[12] M. S. Medlicott and S. R. Harris, "A systematic review of the effectiveness of exercise, manual therapy, electrotherapy, relaxation training, and biofeedback in the management of temporomandibular disorder," *Physical Therapy*, vol. 86, no. 7, pp. 955–973, 2006.

[13] J. L. Crow, N. B. Lincoln, F. M. Nouri, and W. de Weerdt, "The effectiveness of EMG biofeedback in the treatment of arm function after stroke," *International Disability Studies*, vol. 11, no. 4, pp. 155–160, 1989.

[14] K. Sartipi, N. P. Archer, and M. H. Yarmand, "Challenges in developing effective clinical decision support systems," in *Efficient Decision Support Systems—Practice and Challenges in Biomedical Related Domain*, chapter 1, InTech, Rijeka, Croatia, 2011.

[15] S. L. Dimmick, C. Mustaleski, S. G. Burgiss, and T. Welsh, "A case study of benefits & potential savings in rural home telemedicine," *Home Healthcare Nurse*, vol. 18, no. 2, pp. 124–135, 2000.

[16] P. G. Clark, S. J. Dawson, C. Scheideman-Miller, and M. L. Post, "Telerehab: stroke teletherapy and management using two-way interactive video," *Neurology Report*, vol. 26, no. 2, pp. 87–93, 2002.

[17] L. Liu and M. Miyazaki, "Telerehabilitation at the University of Alberta," *Journal of Telemedicine and Telecare*, vol. 6, supplement 2, pp. 47–49, 2000.

[18] T. G. Russell, G. A. Jull, and R. Wootton, "The diagnostic reliability of Internet-based observational kinematic gait analysis," *Journal of Telemedicine and Telecare*, vol. 9, supplement 2, pp. S48–S51, 2003.

[19] M. J. Rosen, "Telerehabilitation," *NeuroRehabilitation*, vol. 12, no. 1, pp. 11–26, 1999.

[20] V. G. Popescu, G. C. Burdea, M. Bouzit, and V. R. Hentz, "A virtual-reality-based telerehabilitation system with force feedback," *IEEE Transactions on Information Technology in Biomedicine*, vol. 4, no. 1, pp. 45–51, 2000.

[21] D. J. Reinkensmeyer, C. T. Pang, J. A. Nessler, and C. C. Painter, "Web-based telerehabilitation for the upper extremity after stroke," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 10, no. 2, pp. 102–108, 2002.

[22] M. K. Holden, T. A. Dyar, and L. Dayan-Cimadoro, "Telerehabilitation using a virtual environment improves upper extremity function in patients with stroke," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 1, pp. 36–42, 2007.

[23] N. Sultan, "Making use of cloud computing for healthcare provision: opportunities and challenges," *International Journal of Information Management*, vol. 34, no. 2, pp. 177–184, 2014.

[24] H. Wu, Q. Wang, and K. Wolter, "Mobile healthcare systems with multi-cloud offloading," in *Proceedings of the 14th IEEE International Conference on Mobile Data Management (MDM '13)*, vol. 2, pp. 188–193, IEEE, Milan, Italy, June 2013.

[25] W. Liu and E. K. Park, "E-Healthcare cloud computing application solutions: cloud-enabling characteristices, challenges and adaptations," in *Proceedings of the International Conference on Computing, Networking and Communications (ICNC '13)*, pp. 437–443, January 2013.

[26] E. Hendrick, B. Schooley, and C. Gao, "CloudHealth: developing a reliable cloud platform for healthcare applications," in *Proceedings of the IEEE 10th Consumer Communications and Networking Conference (CCNC '13)*, pp. 887–891, January 2013.

[27] C. He, X. Fan, and Y. Li, "Toward ubiquitous healthcare services with a novel efficient cloud platform," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 1, pp. 230–234, 2013.

[28] C. S. Wu and I. Khoury, "e-healthcare web service broker infrastructure in cloud environment," in *Proceedings of the 8th IEEE World Congress on Services (SERVICES '12)*, pp. 317–322, Honolulu, Hawaii, USA, June 2012.

[29] R. Wolski, S. Gurun, C. Krintz, and D. Nurmi, "Using bandwidth data to make computation offloading decisions," in *Proceedings of the IEEE International Symposium on Parallel and Distributed Processing (IPDPS '08)*, pp. 1–8, IEEE, Miami, Fla, USA, April 2008.

[30] K. Karthik, "Cloud computing for mobile users: can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, 2010.

[31] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile Networks and Applications*, vol. 18, no. 1, pp. 129–140, 2013.

[32] H.-T. Pham, J.-J. Kim, T. L. Nguyen, and Y. Won, "3D motion matching algorithm using signature feature descriptor," *Multimedia Tools and Applications*, vol. 74, no. 3, pp. 1125–1136, 2015.

[33] S. Li and P. Pathirana, "A kinematic based evaluation of upper extremity movement smoothness for tele-rehabilitation," in *Inclusive Smart Cities and e-Health*, vol. 9102 of *Lecture Notes in Computer Science*, book section 18, pp. 221–231, Springer International Publishing, 2015.

[34] S. Li, M. Ferraro, T. Caelli, and P. N. Pathirana, "A syntactic two-component encoding model for the trajectories of human actions," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 6, pp. 1903–1914, 2014.

[35] Y. S. Lee, P. N. Pathirana, R. J. Evans, and C. L. Steinfort, "Noncontact detection and analysis of respiratory function using microwave Doppler radar," *Journal of Sensors*, vol. 2015, Article ID 548136, 13 pages, 2015.

[36] S. W. Ekanayake, A. J. Morris, M. Forrester, and P. N. Pathirana, "Biokin: an ambulatory platform for gait kinematic and feature assessment," *Healthcare Technology Letters*, vol. 2, no. 1, pp. 40–45, 2015.

[37] S. Mumbaikar and P. Padiya, "Web services based on soap and rest principles," *International Journal of Scientific and Research Publications*, vol. 3, no. 5, 2013.

[38] N. Serrano, J. Hernantes, and G. Gallardo, "Service-oriented architecture and legacy systems," *IEEE Software*, vol. 31, no. 5, pp. 15–19, 2014.

[39] F. AlShahwan and K. Moessner, "Providing SOAP web services and RESTful web services from mobile hosts," in *Proceedings of the 5th International Conference on Internet and Web Applications and Services (ICIW '10)*, pp. 174–179, Barcelona, Spain, May 2010.

[40] D. Box, D. Ehnebuske, G. Kakivaya et al., "Simple object access protocol (soap) 1.1," W3C Note, 2000.

[41] R. Fielding, "Representational state transfer," in *Architectural Styles and the Design of Netowork-based Software Architecture*, pp. 76–85, 2000.

[42] C. Amon and F. Fuhrmann, "Evaluation of the spatial resolution accuracy of the face tracking system for kinect for windows v1 and v2," in *Proceedings of the 6th Congress of the Alps Adria Acoustics Association*, Graz, Austria, October 2014.

[43] Biokin, http://biokin.com.au/.

[44] M. S. Karunarathne, S. A. Jones, S. W. Ekanayake, and P. N. Pathirana, "Remote monitoring system enabling cloud technology upon smart phones and inertial sensors for human kinematics," in *Proceedings of the 4th IEEE International Conference on Big Data and Cloud Computing (BdCloud '14)*, pp. 137–142, Sydney, Australia, December 2014.

[45] Y. S. Lee, P. N. Pathirana, C. L. Steinfort, and T. Caelli, "Monitoring and analysis of respiratory patterns using microwave doppler radar," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 2, pp. 1–12, 2014.

[46] H. Demirkan and D. Delen, "Leveraging the capabilities of service-oriented decision support systems: putting analytics and big data in cloud," *Decision Support Systems*, vol. 55, no. 1, pp. 412–421, 2013.

[47] W. Liu and E. K. Park, "e-healthcare security solution framework," in *Proceedings of the 21st International Conference on Computer Communications and Networks (ICCCN '12)*, Munich, Germany, August 2012.

[48] T. Jing, X. Cui, W. Cheng, S. Zhu, and Y. Huo, "Enabling smartphone based HD video chats by cooperative transmissions in CRNs," in *Wireless Algorithms, Systems, and Applications*, vol. 8491 of *Lecture Notes in Computer Science*, pp. 636–647, Springer, 2014.