

Research Article

Framework of Resource Management for Intercloud Computing

Mohammad Aazam and Eui-Nam Huh

Computer Engineering Department, Kyung Hee University, Suwon 446-701, Republic of Korea

Correspondence should be addressed to Eui-Nam Huh; johnhuh@khu.ac.kr

Received 19 June 2014; Revised 18 August 2014; Accepted 18 August 2014; Published 11 September 2014

Academic Editor: Jong-Hyuk Park

Copyright © 2014 M. Aazam and E.-N. Huh. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

There has been a very rapid increase in digital media content, due to which media cloud is gaining importance. Cloud computing paradigm provides management of resources and helps create extended portfolio of services. Through cloud computing, not only are services managed more efficiently, but also service discovery is made possible. To handle rapid increase in the content, media cloud plays a very vital role. But it is not possible for standalone clouds to handle everything with the increasing user demands. For scalability and better service provisioning, at times, clouds have to communicate with other clouds and share their resources. This scenario is called Intercloud computing or cloud federation. The study on Intercloud computing is still in its start. Resource management is one of the key concerns to be addressed in Intercloud computing. Already done studies discuss this issue only in a trivial and simplistic way. In this study, we present a resource management model, keeping in view different types of services, different customer types, customer characteristic, pricing, and refunding. The presented framework was implemented using Java and NetBeans 8.0 and evaluated using CloudSim 3.0.3 toolkit. Presented results and their discussion validate our model and its efficiency.

1. Introduction

Digital media have convincingly surpassed traditional media, as a result of which this trend makes big and possibly long-term changes to the content being exchanged over the Internet. The global Internet video traffic had surpassed global peer-to-peer (P2P) traffic in 2010 [1]. Excluding the amount of video exchanged through P2P file sharing, at the time being, Internet video is 40 percent of consumer Internet traffic. Since 2012, it has surpassed the mark of 50 percent and will reach 62 percent by the end of 2015. If all forms of videos are counted, the number will be approximately 90 percent by 2015 [2]. To meet the great opportunities and challenges coming along with media revolution, sophisticated technology and better facilities with more powerful capabilities have become the most urgent demands.

Cloud computing recently has emerged and advanced rapidly as a promising and inevitable technology. Cloud computing platform provides highly scalable, manageable, and schedulable virtual servers, storage, computing power, virtual networks, and network bandwidth, according to user's requirement and affordability [3, 4]. Therefore, it can provide solution package for the media revolution, if wisely designed

for media cloud and deployed and integrated with the advanced technologies on media processing, transmission, and storage, keeping in view the industrial and commercial trends and models as well [5]. An average user generates content very quickly, until he/she runs out of storage space. Most of the content may be used frequently by the user, which requires to be accessed easily. Media management is among the key aspects of cloud computing, since cloud makes it possible to store, manage, and share large amounts of digital media.

Cloud computing is a handy solution for processing content in distributed environments. Cloud computing provides ubiquitous access to the content [6], without the hassle of keeping large storage and computing devices. Sharing large amount of media content is another feature that cloud computing provides [7]. Other than social media, traditional cloud computing provides additional features of collaboration and editing of content. Also, if content is to be shared, downloading individual files one by one is not easy. Cloud computing caters to this issue, since all the content can be accessed at once by other parties, with whom the content is being shared.

The increasing demands in cloud computing arena has resulted in more heterogeneous infrastructure, making interoperability an area of concern. Due to this, it becomes a challenge for cloud customers to select appropriate cloud service provider (CSP) and hence it ties them to a particular CSP. This is where intercloud computing comes into play. Although intercloud computing is still in its infancy, its purpose is to allow smooth interoperability between clouds, regardless of their underlying infrastructure. This allows users to migrate their workloads across clouds easily. Cloud brokerage is a promising aspect of intercloud computing [8, 9].

Most of the data-intensive applications are now deployed on the clouds. These applications, storage, and data resource are so diversely located that they have to reach even cross-continental networks. Due to this, performance degradation in networks affects the performance of cloud systems and user requests. To ensure service quality, especially for bulk-data transfer, resource reservation and utilization become a critical issue [10].

Previous works mainly focus on integrated and collaborative uses of resources to meet application requirements. They do not focus on bulk-data transfer consistency and efficiency. They assume that all resources are connected by high-speed stable networks. Continuously growing cloud market faces new challenges now. Even though users have well collaborated end systems and resources are allocated according to their needs, still, bulk-data transfer for cross-continental users in remote places might create performance bottleneck. For instance, multimedia services like IP Television (IPTV) rely on availability of sufficient network resources and hence they have to be operated within the limitation of time constraints [10–12].

The findings in the study presented by Deelman et al. support the fact that right amount of resource allocation can reduce significant amount of cost, without affecting the performance [13]. In user's perspective, fairness is a key concern in resource allocation and pricing. Current pay-as-you-go billing mechanisms typically charge the users on hourly basis, which incurs unfairness [14]. As cloud computing is an economically driven paradigm, fairness becomes an important feature in a pricing scheme. In terms of economics, pricing fairness has two types: personal fairness and social fairness. Personal fairness is subjective and means that the pricing should be affordable for the users. Social fairness, on the other hand, means how much fairness was adopted among the users, using the same service. Having the same cost for the same services utilized is known as social fairness. An unfair pricing develops dissatisfaction in users and eventually the service provider loses its customers [14].

Cloud computing still faces some open challenges, but to provide better reliability, availability, cost-efficiency, and QoS, intercloud computing has already been envisioned. Research on intercloud computing is still not mature enough, but its effectiveness cannot be denied by any means [15]. CSPs have their customers dispersed all around the globe. To serve them optimally, CSPs have to set up many of their data centers at different geographical locations. Existing systems are not capable enough of coordinating dynamically the load

distribution among data centers to determine optimal location for hosting services to achieve desired performance. Furthermore, users' geographical distribution cannot be predicted as well. Hence, load coordination as well as service distribution has to be done automatically. Intercloud computing is meant to counter this problem. It provides scalable provisioning of services with consistent performance, under variable workload and dynamically changing requirements. It supports dynamic expansion and contraction of resources to handle abrupt variations in service demands [9]. Broker's responsibility is to identify appropriate CSP, according to the needs of its customer, through cloud exchange. Broker negotiates with the gateway to allocate resources, according to user and service requirements [16].

When there is no broker and user interaction with the CSP directly, user has to decide what and how many instances to be reserved. When there is an intermediary (broker) between CSP and the user, then broker asks its users to decide on resource reservation and helps them with this decision making [17]. With cloud federation, where multiple independent cloud service providers cooperate with each other, a customer's request from one CSP is entertained by another CSP through the mediation of an intermediary, termed as cloud broker [18]. The broker has to manage resource allocation, which can be tricky at times. This resource allocation can be ad hoc or advanced. But broker has to intelligently decide what to do based on customer's behavior/characteristic and type and price of service(s) [19]. For some services, broker performs ad hoc reservation of resources, while, in some cases, advanced reservation of resources is to be done. In this case, broker has to decide what resources are to be reserved and to what extent [19]. CSPs prefer to have the information about resource reservation in advance. This allows them to adapt to the demands of service and user and allows them to do capacity planning in a better way [17]. For on-demand reservation of resources, CSPs have to manage everything at the time the demand has been made. This makes management of resources difficult for the CSPs and hence reservation is more costly for the users.

There is no complete brokerage model present so far that could handle all the important tasks, from resource prediction, reservation, billing, and refunding. For example, Amazon Elastic Compute Cloud (EC2) charges its users on hourly basis. Thus, if a resource is used for few minutes, it would still charge the user on hourly basis, as if that instance was running for an hour. This is certainly not appreciated by the users. This becomes more of an issue when billing cycle is even longer, like, daily basis (e.g., VPS.NET [20]) [17]. Since the demands are dynamic and may change at any time, it remains a challenge for the broker to decide what amount of resources has to be reserved on-demand [17]. In this paper, we address this issue by presenting a resource management model for intercloud broker.

In rest of the paper, Section 2 is on the already done works in this regard and their shortcomings. Section 3 explains intercloud broker. Section 4 presents broker's resource management model. Section 5 is on performance evaluation. Section 6 concludes the paper.

2. Related Work

Intercloud computing is still in its start due to which there is no standard architecture available for data communication, media storage, compression, and media delivery. Already done studies mainly focus on presenting architectural blueprints for this purpose. In Intel-HP Viewpoint paper [21], industrial overview of the media cloud is presented. The authors state that media cloud is the solution to suffice the dramatically increasing trends of media content and media consumption. For media content delivery, Quality of Service (QoS) is going to be the main concern. We discuss it in detail by presenting end to end QoS provisioning mechanism using flow label of IPv6 and multiprotocol label switching (MPLS) in [22] and impact of IPv6 tunneling in clouds in [23]. To reduce delay and jitter of media streaming, better QoS is required, for which [24] proposes media-edge cloud (MEC) architecture. The authors present in this paper that an MEC is a cloudlet which locates at the edge of the cloud. MEC is composed of storage space, central processing unit (CPU), and graphics processing unit (GPU) clusters. The MEC stores, processes, and transmits media content at the edge, thus incurring a shorter delay. In turn the media cloud is composed of MECs, which can be managed in a centralized or peer-to-peer (P2P) way.

Ferretti et al. [25] present an approach to use a pair of proxies, a client proxy at the user's side and a server proxy at the cloud side, to integrate the cloud seamlessly to the wireless applications. Diaz-Sanchez et al. [26] and Díaz-Sánchez et al. [27] also present proxy as a bridge, for sharing the contents of home cloud to other home clouds and to the outside, public media clouds. This proxy can do additional task of indexing the multimedia content, allowing public cloud to build search database and content classification. Media cloud can then provide discovery service to the users to search the content of their interest. Huang et al. [28] also present a proxy scheme for transcoding and delivery of media. On the other hand, Jin and Kwok [29] propose usage of P2P for delivering media stream outside the media cloud. In both cases, it builds a hybrid architecture, which includes P2P as well as media cloud.

Transcoding and compression of media content require a lot of resources. Pereira et al. present an architecture in [30, 31], in which MapReduce model is applied for this purpose, in private and public clouds.

Feng et al. [32] have proposed the concept of stream-oriented cloud and stream-oriented object. The authors introduce stream-oriented cloud with a high-level description. Mahajan et al. [33], Lim and Lee [34], and Ahmed et al. [35] discuss load balancing among virtual machines and applications.

Rogers and Cliff present [19] a resource allocation mechanism, but resource prediction and detailed billing, along with refunding issue, are not considered (see Algorithms 1 and 2). Ki-Woong et al. [36] present a billing system with some security features. To resolve different types of disputes in the future, a mutually verifiable billing system is presented. Their work only focuses on the reliability of transactions made in purchasing and consuming resources. They do not focus

on the overall resource management, pricing, refunding, or similar important features of cloud broker.

Wang et al. [17] propose a brokerage service for reservation of instances. The authors propose a brokerage service for on-demand reservation of resources, for IaaS clouds. Their work is limited to only on-demand jobs and they do not present anything beyond that. Jrad et al. present a generic architecture of broker. They present how broker handles service level agreement (SLA) management and interoperability of resources [8, 9]. Yang et al. present resource allocation algorithm in a simplistic way [10]. Deelman et al. present performance tradeoffs of different resource provisioning plans. They also present tradeoffs in terms of storage fee of Amazon S3 [13]. Their work does not take into account pricing strategies and other resource management tasks. Ibrahim et al. present the concept of fairness in pricing in respect of microeconomics [14]. Grozev and Buyya present basic taxonomies for intercloud architecture [15], not discussing the way broker handles resources, services, and customers. Rajkumar et al. present architectural fundamental of intercloud computing [16]. Villegas et al. present [18] how multiple clouds are influenced by creating a cloud federation environment. Their work also lacks any further discussion on brokerage.

3. Intercloud Broker

As discussed different types of media content are rapidly increasing and so are users' demands. A single cloud cannot always fulfill the requests or provide required services. There comes a situation when two or more clouds have to communicate with each other, or another intermediary comes into play and federates the resources of two or more clouds. In intercloud terminology, that intermediary is known as "cloud broker" or simply "broker." Broker is the entity which introduces the cloud service customer (CSC) to the cloud service provider (CSP) and vice versa.

Cloud broker provides a single interface through which multiple clouds can be managed and share resources. Cloud broker operates outside of the clouds and controls and monitors those clouds. The main purpose of the broker is assisting the customer to find the best provider and service, according to customer's needs, with respect to specified SLA and providing the customer with a uniform interface to manage and observe the deployed services. Cloud broker earns its profit by fulfilling requirements of both parties. Cloud broker uses a variety of methods, such as a repository for data sharing and integration across data sharing services, to develop a commendable service environment and achieve the best possible deal and SLA between two parties (i.e., CSP and CSC) [30]. Broker typically makes profit either by taking remuneration from the completed deal or by varying the broker's spread or some combination of both [37]. The spread is the difference between the price at which a broker buys from seller (provider) and the price at which it sells to the buyer (customer).

4. Broker's Resource Management Framework

One of the cloud computing core attributes is pay-as-you-go billing model. It enables the customers to scale their capacity

```

(1)  $totalUsers = n$ 
(2) for all users  $i$  do
     $relProb[i] \leftarrow P_i(L | H)$ 
     $userCh[i] \leftarrow \mu_i$ 
(3) Input  $pCost[i]$  for service  $s$ 
(4)  $rRes[i] \leftarrow c * d$ 
     $c \leftarrow (pCost[i] * relProb[i] * n)$ 
     $d \leftarrow (1 - userCh[i])$  for  $D_s$ 
(5) end for
(6)  $thresh \leftarrow \theta$ 
(7) for all users  $i$  and  $relProbk$  do
     $probSum += relProb[i]$ 
     $avgRelProb \leftarrow probSum/k$ 
(8) end for
(9) if  $thresh > avgRelProb$ 
     $rAlloc$ 
(10) end if

```

ALGORITHM 1: Resource estimation.

```

(1) for all  $relProb == l$  do
     $userChL \leftarrow \frac{pCost - relProbL}{pCost * (1 - relProbL)}$ 
     $totalPriceL \leftarrow (pCost + userChL + userCh)$ 
(2) end for
(3) for all  $relProb == h$  do
     $userChH \leftarrow \frac{pCost - relProbH}{pCost * (1 - relProbH)}$ 
     $totalPriceH \leftarrow (pCost + userChH + userCh)$ 
(4) end for

```

ALGORITHM 2: Cost estimation.

according to their changing requirements and pay for the consumed resources.

CSCs contact broker to acquire the required service(s) at best price. Broker performs the negotiation and SLA tasks with CSP. Once the contract is settled, the service is provided to the customer. In this regard, not only does broker provide services on ad hoc basis, but also it has to predict consumption of resources, so that they can be allocated in advance, allowing more efficiency and fairness at the time of consumption. This prediction as well as preallocation of resources also depends upon user's behavior and its probability of using those resources in future.

To handle commercial services, broker has a cost management system. Broker includes application programming interfaces (APIs) and a standard abstract API, which is used to manage cloud resources from different cloud providers. Broker holds another abstract API for the negotiation of cloud service facilities with the customer. Different modules perform a specific task in broker's architecture; for example, registration of new services is handled by service registration manager. Deployment of services and making them available is done by deployment manager. Similarly, each module has its own specific utility.

For this intermediary service, broker performs pricing and billing, which is presented in this section.

We formulate the estimation of required service as

$$R_{res} = \sum_{i=0}^n \begin{cases} (C_{m_i} * n * P_i(L | H)) * (1 - \mu_i) * D_s \\ 0, \end{cases} \quad (1)$$

where R_{res} represents required resources mapped value, C_m is the maximum cost a particular user can afford or is willing to pay, n is the number of cloud customers, and $P(L | H)$ is the probability of a particular customer of giving up resources. For simplicity, we have categorized it into two probabilities, low (L) or high (H) probability. Consider

$$\begin{aligned} 0 > L &\leq 0.5, \\ 0.5 > H &\leq 1, \end{aligned} \quad (2)$$

where " μ " is user characteristic, a constant decision variable value, which is assigned by the broker to each user, according to its characteristic or history and D_s is duration for which a particular service has been requested.

With this formulation, cloud broker can determine future resource requirements. It is important for cloud broker to

rightly decide when to reserve resources and not to waste precious cloud resources. It will also help power consumption management, which is becoming a concern for cloud data-centers.

After predicting the future resource requirement, next task is price allocation and billing. Pricing is not a straight forward matter. There are different pricing strategies available.

Time-Based Pricing. A dynamic pricing strategy, based on a large number of data gathered from customers about their behavior and trends.

Value-Based Pricing. Pricing a service on the basis of the value it holds for the customer, instead of the cost of production.

Target-Pricing. Selling the services at a planned profit rate.

Psychological Pricing. Selling at a rate having positive psychological impact; for example, a service of \$10 is priced at \$9.99 or \$9.95.

Price Discrimination. Setting up different prices in different segments of markets, based on class, age, behavior, and so forth.

Premium Pricing. Artificially setting up the prices higher. This practice exploits customers' tendency of having perception that higher priced products are of better quality.

We formulate broker's generic pricing method as follows:

$$\rho_{S_{P(L)}} = \int_0^t C_m + \left(\frac{\mu_L}{D_s}\right) + \left(\frac{\mu}{D_s}\right), \quad (3)$$

$$\rho_{S_{P(H)}} = \int_0^t C_m + \left(\frac{\mu_H}{D_s}\right) + \left(\frac{\mu}{D_s}\right),$$

where $\rho_{S_{P(L)}}$ is the price for service S, which has been requested with relinquish probability P_L . Similarly, $\rho_{S_{P(H)}}$ is the price with relinquish probability P_H , μ_L is the decision variable for user P_L , calculated through (4), and μ_H is the decision variable for user P_H , calculated through (5). Consider

$$\mu_L = \frac{C_m * P_L}{\delta}, \quad (4)$$

$$\mu_H = \frac{C_m * P_H}{\delta}, \quad (5)$$

where δ represents average profit earned so far from the currently requesting CSC.

When n number of customers have requested a service S, broker has to decide whether or not to register the service. Based on the relinquish probabilities of each customer, broker makes this decision. For every service, depending upon the type, duration, and cost of service, broker sets a threshold value " θ ". It then accumulates the relinquish probabilities of each subscriber (P_T). If the accumulated probability is greater than or equal to the threshold value, it registers the service; otherwise, the service is not provided, because with some services, a particular minimum number of customers are required to register. Otherwise, that service becomes very

costly, either not affordable for the customer(s) or leaving broker and service provider with a very low margin of profit. This task is formulated as follows:

$$P_T = \bar{x} \left(\sum_{i=0}^n P(L | H)_i \right). \quad (6)$$

Service is provided if $P_T > \theta$.

When the service is being utilized, customer can decide to discontinue at any stage. At that time, broker has to halt the service and refund the remaining amount to the customer. In this case, broker has to take into account the utilized resources, or consumed services, and the remaining service value of the decided total initial service. This can be formulated through the following equations:

$$Y_t = Y_{un} + Y_{deg}, \quad (7)$$

$$R_{un} = 1 - \frac{\alpha}{100}, \quad (8)$$

where Y_t is the total amount to be refunded, Y_{un} is the refund amount of unutilized resources, and Y_{deg} is the refund amount to be paid on quality degradation. During service delivery, it is not always possible to deliver the service exactly according to the promise made during SLA. Y_{un} is calculated through (9) to (12) and Y_{deg} is further calculated through (13).

In (8), R_{un} represents unutilized resources, while α represents utilized resources.

For those customers who have used more service, broker and service provider have earned more money from them. Therefore, when they quit the service, they can be provided with some appreciation amount Y_{unA} while refunding. We call it appreciation index ω . For example, customers who have used 60% or more of the service are eligible for this. In that case, the refund amount should linearly increase, encouraging the customer, which in turn works as an appealing factor and allows customer to return to that service provider again and again and consume more services. In this case, the formula would be

$$Y_{unA} = R_{un} * C_m + \omega, \quad (9)$$

$$\omega = \log \alpha, \quad (10)$$

$$Y_{unD} = R_{un} * C_m + \varepsilon, \quad (11)$$

$$\varepsilon = \ln \left(\frac{\alpha}{100} \right). \quad (12)$$

ε is the depreciation index, which deducts some amount, based on business policy, from those customers who used fewer services. Currently in our model, we apply depreciation index when resource utilization is less than 60%:

$$Y_{deg} = \left(\frac{Q_{SLA}}{Q_a} * \frac{\beta}{100} \right) * (C_m - Y_{un}), \quad (13)$$

where Q_a is the acquired QoS, Q_{SLA} is the promised QoS, during SLA, and β is the ratio of refund amount, set by the broker, based on business contract and condition (e.g., 10% of the total amount).

TABLE 1: Simulation setup.

System	Intel(R) Core(TM) i5-M430, 2.27 GHz
Memory	4 GB
Simulator	CloudSim 3.0.3
Operating system (OS)	Window 7 Home Premium

TABLE 2: Parameters' setting for evaluation.

Parameters	Range
Service level agreement (Q_{SLA})	0.9
Acquired service quality (Q_a)	0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1
Service price (ρ)	100, 150, 200, 250, ..., 1000
Service utilization	40%, 50%, 60%, 70%, 80%, 90%
Refund (Υ)	30%~90%
SLA based refund (Υ_{deg})	0.8~0.1
Number of registered services	10
Service duration (D_s) in months	1, 2, 3, 4, 5, 6

5. Efficiency and Performance Evaluation

In this section, we present efficiency of our proposed refundable service model. We defined our service model through authentic algorithm to evaluate the effectiveness in cloud computing business. Our main objective is to observe the influence of performance factors on the systems and test the feasibility of our method.

5.1. Experimental Setup. The experiment environment for our model's evaluation is shown in Table 1. We have considered different parameters to estimate the required resources, pricing, and billing for two types of users and refund amount calculation. Table 2 shows the parameters' setting.

5.2. Resource Prediction according to User Type for Different Services. When different CSCs are requesting a particular service, the CSP or broker has to analyze what number of resources has to be allocated for that service, based on the type of customer. For CSCs having low relinquishing probability, priority in resource allocation is given. Figure 1 shows the unit of resources being predicted for both types of customers, for all types of registered services.

The unit is greater for L customers, while it is smaller for H customers, because of their behavior. Since there are more chances of an H customer to relinquish the service(s), as a result, more priorities and quality are provided to the more loyal customer, having L probability. Resources increase as the service price increases.

5.3. Service Price Based on Customer's Characteristic. According to the cloud resource consumer's characteristic, service price might be different. Those users who have a trend to give up their services, their final service price would be higher, compared to the other types of customers who always utilized

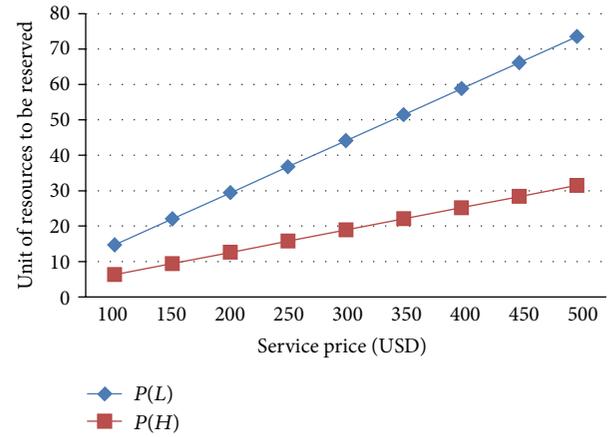


FIGURE 1: Resource prediction for different types of CRCs, for different requested services.

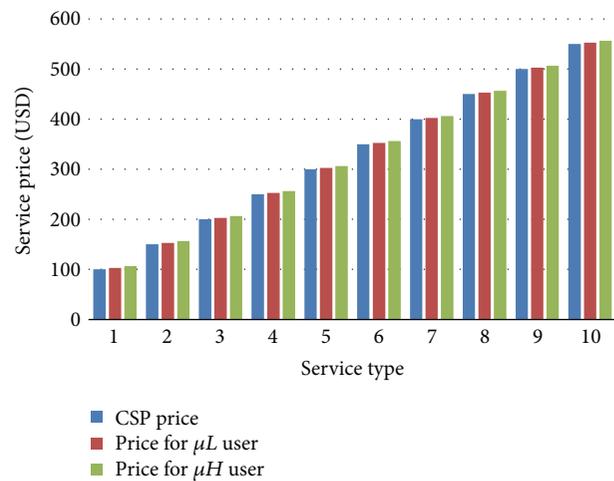


FIGURE 2: Service price based on user's characteristic.

their reserved resources. From Figure 2, we can see the actual service price from CSP is USD 100. However, users L (utilizing most of their reserved services) need to pay around USD 102, since they will get some sort of advantage (e.g., refundable service) from cloud broker. In addition, the users H (higher tendency to give up their service) need to pay around USD 106, because cloud broker has enough risk to pay back for unutilized resources. Therefore, those users are assigned a greater amount of the service price. Similarly, prices vary for both types of users, for each service ranging from USD 100 to USD 400. In every case, H users' prices have always been greater than the actual price as well as L price.

5.4. Refund Amount Based on Service Utilization. In this part of evaluation, reserved services' prices range from USD 100 to USD 500 and service utilization ranges from 50% to 90% of total reservation. Service quality is kept constant here at 0.9 (SLA fulfilled).

Figure 3 shows refund amount for the service reserved, on the basis of resources consumed. As the service price

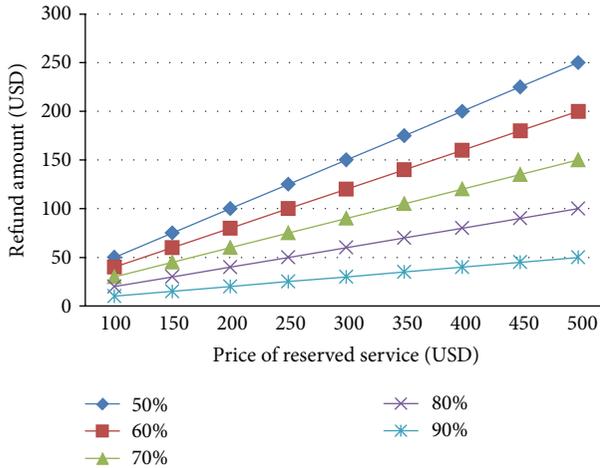


FIGURE 3: Reserved service price and refund amount.

increases, the refund amount also increases linearly. Similarly, the more the resource consumption is, the less the refund amount to be paid back to the customer would be. The graph shows that, for a USD 100 service, refund amount decreases on the basis of the amount of resources consumed. For 50% resource consumption, refund amount is higher, for 60%, it decreases comparatively, and so on. Similarly, for a more expensive service, the refund amount is increasing.

5.5. Appreciated Refund Amount for Different Resource Consumption with Fixed Service Quality and Resource Reservation. In this part, service utilization ranges from 30% to 90%. With fixed resource reservation and reasonable service quality (0.7-0.8 achieved SLA), impact on refund amount is measured. When cloud broker and CSP reserve resources or provide services to the CRC, they expect that the service would be completely utilized. Otherwise, they have to pay the customer back the remaining amount, which is a kind of loss for them. In this case, CSP and broker have to appreciate those customers who utilized more of their subscription and depreciate those who did not. In this part, we have set the level of decision making for appreciation or depreciation at 60% resource consumption. If a CRC consumes less than 60% of its reserved resources, it will get a depreciated refund amount. If it consumes 60% or more, it will receive appreciated refund amount.

As shown in Figure 4, CSC consuming 30% resources of its reserved service which values USD 100 will receive around USD 68, instead of USD 70. This is because the utilized service is less than 60%, since it was depreciated. Similarly, for 40% resource consumption, refund amount is USD 59. On the other hand, for 60% resource consumption, refund amount is around USD 42, instead of USD 40, since it has been appreciated. Similarly, for 90% resource consumption, refund is around USD 12, instead of USD 10. By this, broker motivates its customers to consume more resources or complete the subscription.

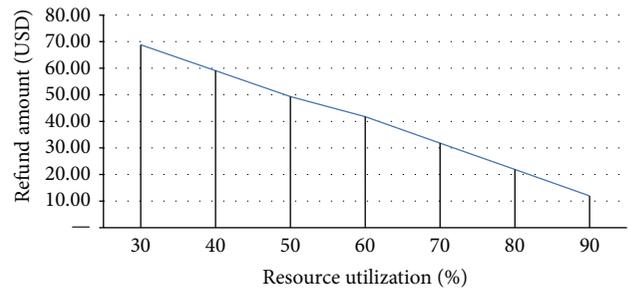


FIGURE 4: Appreciated refund amount according to resource utilization.

6. Conclusion and Future Work

Intercloud computing as well as brokerage is still in its infancy. With rapidly increasing multimedia content in the cloud, QoS, efficiency, and user's satisfaction are becoming a crucial task. The future is of intercloud computing, in which two or more clouds would be interacting with each other for resource management and service provisioning. Brokerage is one of the key aspects of intercloud computing. Broker has to deal with all types of resource management and service availability tasks for its customers. In this case, a complete model that takes care of efficient resource allocation, pricing, billing, and refunding is not yet available. We have tried our best to put all these things under one umbrella and presented a framework. Based on 6 parameters (resource prediction according to user characteristic, pricing according to user characteristic, pricing according to service type, refund according to utilization, appreciated refund, and refund according to service type), the model is evaluated and results are presented. The tested evaluation of this model shows the validity and efficient performance of our proposed model. We intend to extend this part now and work on more varied parameters, under more heterogeneous environment, where different types of devices are being used by customers and diverse services are requested.

Conflict of Interests

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

Acknowledgments

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2014 (H0301-14-1020)) supervised by the NIPA (National IT Industry Promotion Agency). The corresponding author is Professor Eui-Nam Huh. This research was also supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (no. NRF-2013R1A1A2013620). The corresponding author is Professor Eui-Nam Huh.

References

- [1] M. Tan and X. Su, "Media cloud: when media revolution meets rise of cloud computing," in *Proceedings of the 6th IEEE International Symposium on Service-Oriented System Engineering (SOSE '11)*, pp. 251–261, Irvine, Calif, USA, December 2011.
- [2] "Cisco-White-Paper," Cisco Visual Networking Index—Forecast and Methodology, 2010–2015, June 2011.
- [3] S.-H. Lee and I.-Y. Lee, "A secure index management scheme for providing data sharing in cloud storage," *Journal of Information Processing Systems*, vol. 9, no. 2, pp. 287–300, 2013.
- [4] K.-C. Huang, T.-C. Huang, M.-J. T. H.-Y. Chang, and Y.-H. Tung, "Moldable job scheduling for HPC as a service with application speedup model and execution time information," *Journal of Convergence*, vol. 4, no. 4, 2013.
- [5] J. C. Tsai and N. Y. Yen, "Cloud-empowered multimedia service: an automatic video storytelling tool," *Journal of Convergence*, vol. 4, no. 3, pp. 13–19, 2013.
- [6] J. W. K. Gnanaraj and K. E. E. B. Rajsingh, "Smart card based time efficient authentication scheme for global grid computing," *Human-centric Computing and Information Sciences*, vol. 3, article 16, 2013.
- [7] K. Chorianopoulos, "Collective intelligence within web video," *Human-Centric Computing and Information Sciences*, vol. 3, p. 10, 2013.
- [8] F. Jrad, J. Tao, and A. Streit, "SLA based service brokering in intercloud environments," in *Proceedings of the 2nd International Conference on Cloud Computing and Services Science (CLOSER '12)*, pp. 76–81, Porto, Portugal, April 2012.
- [9] F. Jrad, J. Tao, and A. Streit, "Simulation-based evaluation of an intercloud service broker," in *Proceedings of the 3rd International Conference on Cloud Computing, GRIDs, and Virtualization*, Nice, France, 2012.
- [10] Y. Yang, Y. Zhou, L. Liang, D. He, and Z. Sun, "A service-oriented broker for bulk data transfer in cloud computing," in *Proceedings of the 9th International Conference on Grid and Cloud Computing (GCC '10)*, pp. 264–269, Nanjing, China, November 2010.
- [11] G. Christou, "A comparison between experienced and inexperienced video game players' perceptions," *Human-Centric Computing and Information Sciences*, vol. 3, article 15, 2013.
- [12] H. Yo-Sung, "Challenging technical issues of 3D video processing," *Journal of Convergence*, vol. 4, no. 1, 2013.
- [13] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good, "The cost of doing science on the cloud: the montage example," in *Proceedings of the ACM/IEEE Conference on Supercomputing*, IEEE Press, November 2008.
- [14] S. Ibrahim, B. He, and H. Jin, "Towards pay-as-you-consume cloud computing," in *Proceedings of the IEEE International Conference on Services Computing (SCC '11)*, pp. 370–377, Washington, DC, USA, July 2011.
- [15] N. Grozev and R. Buyya, "Inter-Cloud architectures and application brokering: taxonomy and survey," *Software: Practice and Experience*, vol. 44, no. 3, pp. 369–390, 2014.
- [16] B. Rajkumar, R. Ranjan, and R. N. Calheiros, "Intercloud: utility-oriented federation of cloud computing environments for scaling of application services," in *Algorithms and Architectures for Parallel Processing*, Springer, Berlin, Germany, 2010.
- [17] W. Wang, D. Niu, B. C. Li, and B. Liang, "Dynamic cloud resource reservation via cloud brokerage," in *Proceedings of the 33rd IEEE International Conference on Distributed Computing Systems (ICDCS '13)*, pp. 400–409, Philadelphia, Pa, USA, 2013.
- [18] D. Villegas, N. Bobroff, I. Rodero et al., "Cloud federation in a layered service model," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1330–1344, 2012.
- [19] O. Rogers and D. Cliff, "A financial brokerage model for cloud computing," *Journal of Cloud Computing*, vol. 1, no. 1, pp. 1–12, 2012.
- [20] VPS.NET, <http://vps.net>.
- [21] "Moving to the media cloud," Viewpoint Paper, Intel-HP, Palo Alto, Calif, USA, 2010.
- [22] M. Aazam, A. M. Syed, and E.-N. Huh, "Redefining flow label in IPv6 and MPLS headers for end to end QoS in virtual networking for thin client," in *Proceedings of the 19th Asia-Pacific Conference on Communications (APCC '13)*, pp. 585–590, Bali, Indonesia, 2013.
- [23] M. Aazam and E.-N. Huh, "Impact of ipv4-ipv6 coexistence in cloud virtualization environment," *Annals of Telecommunications*, 2013.
- [24] W. Zhu, C. Luo, J. Wang, and S. Li, "Multimedia cloud computing," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 59–69, 2011.
- [25] S. Ferretti, V. Ghini, F. Panzneri, and E. Turrini, "Seamless support of multimedia distributed applications through a cloud," in *Proceedings of the 3rd IEEE International Conference on Cloud Computing (CLOUD '10)*, pp. 548–549, Miami, Fla, USA, July 2010.
- [26] D. Diaz-Sanchez, F. Almenares, A. Marin, and D. Proserpio, "Media cloud: sharing contents in the large," in *Proceedings of the IEEE International Conference on Consumer Electronics (ICCE '11)*, pp. 227–228, Las Vegas, Nev, USA, January 2011.
- [27] D. Díaz-Sánchez, F. Almenarez, A. Marín, D. Proserpio, and P. A. Cabarcos, "Media cloud: an open cloud computing middleware for content management," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 2, pp. 970–978, 2011.
- [28] Z. Huang, C. Mei, L. E. Li, and T. Woo, "CloudStream: delivering high-quality streaming videos through a cloud-based SVC proxy," in *Proceeding of the 30th IEEE INFOCOM*, pp. 201–205, Shanghai, China, April 2011.
- [29] X. Jin and Y.-K. Kwok, "Cloud assisted P2P media streaming for bandwidth constrained mobile subscribers," in *Proceedings of the 16th IEEE International Conference on Parallel and Distributed Systems (ICPADS '10)*, pp. 800–805, Shanghai, China, December 2010.
- [30] R. Pereira, M. Azambuja, K. Breitman, and M. Endler, "An architecture for distributed high performance video processing in the cloud," in *Proceedings of the 3rd IEEE International Conference on Cloud Computing (CLOUD '10)*, pp. 482–489, Miami, Fla, USA, July 2010.
- [31] R. Pereira and K. Breitman, "A cloud based architecture for improving video compression time efficiency: the split and merge approach," in *Proceedings of the Data Compression Conference (DCC '11)*, p. 471, Snowbird, Utah, USA, March 2011.
- [32] J. Feng, P. Wen, J. Liu, and H. Li, "Elastic stream cloud (ESC): a stream-oriented cloud computing platform for Rich Internet Application," in *Proceedings of the International Conference on High Performance Computing and Simulation (HPCS '10)*, pp. 203–208, Caen, France, July 2010.
- [33] K. Mahajan, A. Makroo, and D. Dahiya, "Round robin with server affinity: a VM load balancing algorithm for cloud based infrastructure," *Journal of Information Processing Systems*, vol. 9, no. 3, p. 379, 2013.

- [34] M. Lim and Y. Lee, "A simulation model of object movement for evaluating the communication load in networked virtual environments," *Journal of Information Processing Systems*, vol. 9, no. 3, pp. 489–498, 2013.
- [35] E. Ahmed, A. Akhunzada, A. Gani, S. H. Hamid, and R. Buyya, "Network-centric performance analysis of runtime application migration in mobile cloud computing," *Simulation Modelling Practice and Theory*, 2014.
- [36] P. Ki-Woong, J. Han, J. W. Chung, and K. H. Park, "THEMIS: a mutually verifiable billing system for the cloud computing environment," *IEEE Transactions on Services Computing*, vol. 6, no. 3, pp. 300–313, 2013.
- [37] M. Aazam and E.-N. Huh, "Inter-cloud architecture and media cloud storage design considerations," in *Proceedings of the 7th IEEE International Conference on Cloud Computing (CLOUD '14)*, Anchorage, Alaska, USA, June -July 2014.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

