

Research Article

A Novel Approach for Optimal Multimedia Data Distribution in Mobile Cloud Computing

Pham Phuoc Hung, Mohammad Aazam, Tien-Dung Nguyen, and Eui-Nam Huh

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

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

Received 20 May 2014; Accepted 3 October 2014; Published 23 October 2014

Academic Editor: Young-Sik Jeong

Copyright © 2014 Pham Phuoc Hung et al. 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.

With the integration of mobile computing and cloud computing, more diverse services can be provided to the users, supporting the mobility feature of mobile computing along with the power of cloud computing. This new paradigm still faces challenges, especially in terms of performance. When it comes to multimedia data communication, thin clients (such as smart phones and tablets) suffer because of performance and power constraints. Previously done studies have trivially addressed this problem. Therefore, in our paper, we present a framework in which thick clients (laptop or desktop computers) are incorporated into mobile cloud paradigm with attention paid to user mobility. Its objective is to optimize the distribution of multimedia content between the cloud and the thin clients. Our work comes up with both numerical analysis and simulation to justify the validity and the effectiveness of the proposal approach.

1. Introduction

In the recent times, handheld devices, like smartphones and tablets, have increased rapidly. Users nowadays prefer using lightweight, easy to carry small devices, instead of using laptops or desktop computers. Gartner Inc. report [1] that smartphone and tablet sale has already surpassed PC sale in the last couple of years. Soon they would be users' preferred devices for Internet access [2]. To adapt to the changing and more demanding mobile environment, computing services have been going through innovation. Now with the advent of cloud computing (CC) [3, 4], mobile users can upload their data and outsource tasks to the cloud for further processing. The emerging trend is known as mobile cloud computing (MCC) [5], in which cloud computing and mobile computing come together to provide data intensive and more powerful computing services, along with mobility.

Being different from static computing, MCC faces many challenges, specially related to processing capabilities, constraints of wireless networks, and expensive cloud access, mobility. Particularly, user mobility might severely degrade Quality of Service (QoS) [6, 7]. A lot of solutions have been discussed already, but performance could not be enhanced.

For example, in [8], the authors present a framework for creating virtual MCC providers, through nearby thin clients to receive multimedia data from Internet. But the limited capacity and bandwidth are still a hurdle in cloud access [9, 10]. To improve thin clients' service, it is comprehensive that thin clients should be coupled with thick clients which come with more generous hardware resources and high speed Internet connection (DSL, Fibre, etc.). Keeping this in mind, we introduce thin-thick collaboration architecture, in which thin clients and thick clients can communicate with each other in order to optimize data distribution, specially multimedia content [11, 12], between cloud customers and cloud providers. In so doing, expected QoS requirements can be met. Moreover, our main concern is effective and efficient delivery of data from individual customers to the cloud network, keeping in view mobility of customers. Extensive evaluation shows that our approach improves data distribution efficiency and performs better than the existing approaches.

We divide our paper into following sections: Section 2 gives literature review on existing research work that relates to ours; Section 3 details the motivation scenario which explains why we have come up with the proposed approach; Section 4 specifies the framework architecture and algorithms;

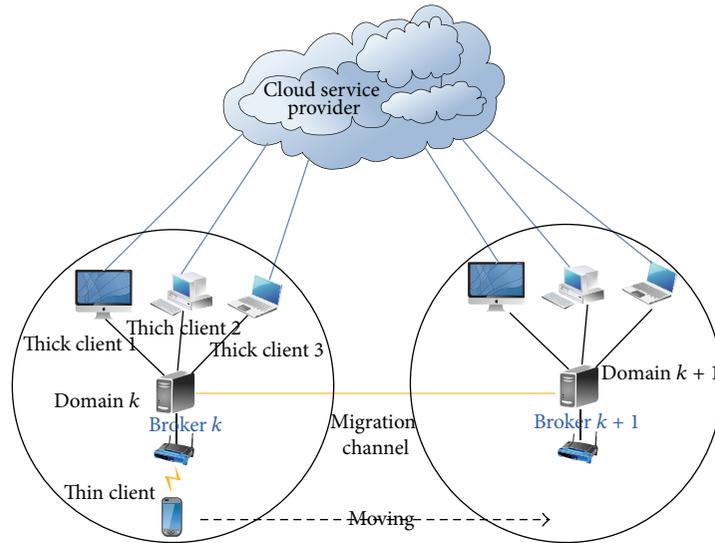


FIGURE 1: Motivating scenario.

Section 5 shows the implementation of our ideas along with performance evaluation; the last section concludes the paper and suggests the future work.

2. Related Work

Research literature has seen efforts in trying to solve the above mentioned problems. In [13], authors propose a new approach for efficient cloud-based synchronization of a number of distributed file system hierarchies. They use a master-slave architecture for propagation of data to reduce traffic. In [14], researchers demonstrate that some resource scheduling techniques can be effective in mitigating the impacts that negatively influence application response time and system utilization. Ross [15] and Kumar and Lu [16] introduce the impact of the data transfer delay on the performance but they do not reckon to use bandwidth efficiently. Similar to our approach, many also tried integrating mobile devices and CC. In [17], Luo suggests a new idea of using cloud to improve mobile device's ability to face the performance challenges for low latency and high throughput of media data processing [12, 18]. Delgado et al. [19] innovate Hyrax that encourages using mobile devices as resource providers in CC platform; yet experiment is not integrated. In [20], the authors just concentrate on using partition policies to hold the effect of application on mobile devices, but they do not solve any other matter related to user mobility [21]. Kim et al. [22] propose an adaptive scheme that optimizes run-time data distribution and collection services for hiding the data transfer time; user mobility is not either. Therefore, in this paper, we try to solve the above shortage by introducing a thin-thick client collaboration to optimize the multimedia data distribution between cloud network and mobile thin clients with attention paid to user mobility.

3. Motivating Scenario

The scenario shown as in Figure 1 exemplifies the utilization of thin-thick client collaboration.

A student wants to watch online movies with her smartphone while she is walking in her university campus. We assume that she is moving at the same speed and cannot use her mobile internet data (as it is slow and has limited capacity). Thus, she has to access her university wireless network. Once connected, her mobile phone (thin client) can send a request (e.g., movie streaming) to the broker which manages several thick clients to serve this request. Each of the chosen thick clients will establish a connection to the corresponding cloud provider which thin client desires to access. Requests are sent to cloud; processed and returned data are delivered to thick clients before gathering at broker and being dispatched to requesting thin clients. It is worth noting that data between cloud and thick client can be delivered in a relatively short time with presumably high-speed Internet connection of these thick clients.

The above situation indicates the possible profit of applying the cooperation between thin clients and thick clients in a classic MCC environment. Such collaboration encouragingly increases the chance of utilizing resources efficiently. However, when mobile users move out of a broker's domain (i.e., no access point connected to the broker is found), all running sessions (called "state") made between clients and cloud providers might be terminated, which leads to service degradation. To sustain the service continuation in this scenario and also to balance the requested workload, it claims the need of using multiple brokers, which means when the mobile clients move to a different domain, the associated broker should be able to follow them by migrating its "state" to the next available broker at the newly visited domain and restore previous session with little impact on service quality. With that in mind, we design our system architecture to deal with the following concerns:

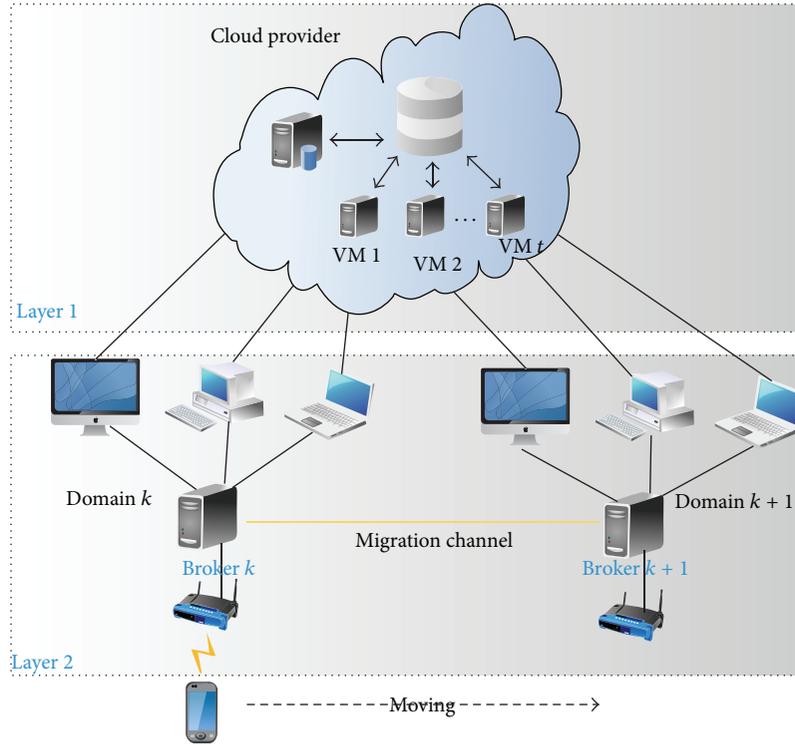


FIGURE 2: System architecture.

- (i) distributing data from cloud providers to thick clients;
- (ii) finding the most appropriate broker to minimize the migration time of “state” from the current broker.

4. System Architecture

The following section gives an insight of our system architecture proposed to address issues discussed in the above scenario.

Different from other approaches that follow a $1/m/1$ model which states that one thin client is served by one server through multiple paths, our system can be viewed as a $1/m/m/m$ model. It expresses that, from a cloud provider (1), chunks of data are sent to multiple thick clients (m) which will then transfer these data to multiple brokers (m) in the scenario. The broker works as the heart of the middle layer, combining and delivering the received data to the intended end user(s) (m). For ease of presentation, our system can be divided into the following two layers as shown in Figure 2.

4.1. Layer 1. In this layer we consider the following: splitting and distributing data from cloud providers to processors (thick clients) with different capacities according to bandwidth of Internet connection. For clarity, we give the important definition and assumption for our system. First, split each block of data into chunks $\{ch_1, ch_2, \dots, ch_n\}$ with different sizes depending on bandwidth. $w(ch_i)$ is the size of a chunk ch_i . b_i is bandwidth from a virtual machine (VM) on a cloud to a processor. Therefore, time spent for the transferring a chunk ch_i from VMs to a processor is $w(ch_i)/b_i$. For

parallelization, the time to transfer chunks to processors should be equal:

$$\frac{w(ch_1)}{b_1} = \frac{w(ch_2)}{b_2} = \dots = \frac{w(ch_i)}{b_i} = t$$

$$\text{Set } S = w(\text{data}) = \sum_{i=0}^n w(ch_i) = t \sum_{i=0}^n b_i \quad (1)$$

$$\text{Thus, } w(ch_i) = t * b_i = \frac{S}{\sum_{i=0}^n b_i} * b_i.$$

According to this value, we can determine the size of each chunk to adapt with bandwidth of each connection. Next step is to sort the processors $\{p_1, p_2, \dots, p_p\}$ depending on their capacities. The processor which has higher capacity will receive big chunks and the processor which has lower capacity will receive small chunks.

4.2. Layer 2. The second layer involves combining the data at brokers and/or overlay services between brokers then transferring it to a thin client. The issue here is how to find the location of the most appropriate broker between the nearby ones to minimize the migration time of “state” from the current broker where a user moves.

Basically, we consider the network topology as the three-tier architecture [23], which consists of core switch (CS), aggregation switch (AgS), and access switch (AcS) as in Figure 3. The number fan-out of CS is denoted by p_2 and $\{a_i \mid i = 1, 2, \dots, p_2\}$ represents transmission delay from CS to corresponding AgSs. The number fan-out and the

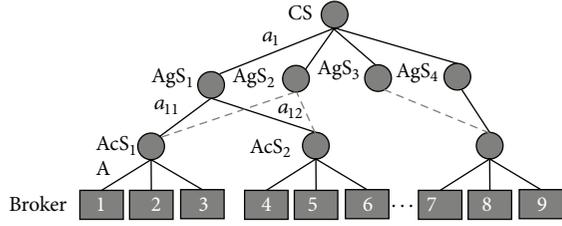


FIGURE 3: Network topology.

transmission delay from AgS to AcS are denoted by p_1 and $\{a_{ij} \mid i = 1, 2, \dots, p_2; j = 1, 2, \dots, p_1\}$, respectively. In the initial state, the transmission delay on each edge between AgS and CS is set equally (i.e., $a_1 = a_2 = \dots = a_{p_2}$).

Similarly, the transmission delay on each edge between AcS and AgS is also set equally (i.e., $a_{11} = a_{12} = \dots = a_{p_2 p_1}$). Let p_0 present the number of fan-out of AcS, and A is the transmission delay between AcS and its brokers. In practical network, it is obvious that transmission delay (T) on an edge of tree is directly proportional to the amount of simultaneously transmitted data “state” on that edge. Hence, the communication cost between brokers depends not only on the number of hop counts but also on the transmission delay which is defined as follows:

$$T(i, j) = \max_{k \in (i, j)} \left(\frac{1}{B_k - l_k} \right), \quad (2)$$

where $T(i, j)$ is the transmission delay from broker i to broker j and B_k and l_k are the bandwidth and the individual load of edge k connecting broker i to broker j (i.e., belongs to path (i, j)), respectively.

Let $H(i, j)$ denote the number of hop counts from broker i to broker j , and $\alpha \in [0, 1]$ is the weighted mean function or the tradeoff number of hop counts and transmission delay between two brokers. Finally, we specify the communication cost $C(i, j)$ from broker i to broker j as follows:

$$C(i, j) = \alpha * H(i, j) + (1 - \alpha) * T(i, j). \quad (3)$$

Therefore,

$$C(i, j) = \begin{cases} 0 & \text{if } i = j \\ \alpha + (1 - \alpha) * A & \text{if } \left\lfloor \frac{i}{p_0} \right\rfloor = \left\lfloor \frac{j}{p_0} \right\rfloor \\ 3\alpha + (1 - \alpha) * \min(A, a_{mx} - l_{mx}, a_{my} - l_{my}) & \text{if } x = \left\lfloor \frac{i}{p_0} \right\rfloor \neq \left\lfloor \frac{j}{p_0} \right\rfloor = y \wedge \left\lfloor \frac{i}{p_0 p_1} \right\rfloor = \left\lfloor \frac{j}{p_0 p_1} \right\rfloor \\ = m \\ 5\alpha + (1 - \alpha) & * \min(A, a_{mx} - l_{mx}, a_m - l_m, a_n - l_n, a_{ny} - l_{ny}) \\ \text{if } m = \left\lfloor \frac{i}{p_0 p_1} \right\rfloor \neq \left\lfloor \frac{j}{p_0 p_1} \right\rfloor = n. \end{cases} \quad (4)$$

```

Input: R//resource demand matrix
          C//communication cost matrix
          sb//source broker
Output: db//destination broker
Function
determineDestBroker(R,C,sb)
begin
    int sum = 0;
    Find group of Brokers (grB)
    close to mobile devices
    min = R(sb,db) * C(sb,grB[0]);
    db = grB[0];
    for (int i = 1; i < sizeof(grB); i++)
    begin
        cost = R(sb,db) * C(sb,grB[i]);
        if (min > cost)
        begin
            min = cost;
            db = grB[i];
        end
    end
    return db;
end

```

ALGORITHM 1: Determine the location of a destination broker.

In addition, we define resource demand matrix $R(i, j)$ as a matrix representing the traffic demand between pairs of brokers (i, j) . Traffic demand comes from the weight of each “state” we want to migrate. Evidently, in order to perfect the system, the pair of brokers which has heavy resource demand should be as close as possible to each other. Thus, the sum of all multiplication of every resource demand $R(i, j)$ and its corresponding communication cost $C(i, j)$ have to be kept minimum. Thence, the objective function can be formulated as follows:

$$\text{Min} \sum_{i, j \in V} (R(i, j)) C(i, j). \quad (5)$$

Our purpose is to minimize this function to find the balance between the lowest cost and the heaviest resource demand between brokers $(i, j) \in V$; V is the set of brokers. It can be understood that the pair of brokers with the lowest communication cost will migrate the heaviest “state” to each other. Hence, we need an efficient algorithm like Algorithm 1 to find the position of destination broker to store the “state” of the current broker to achieve the optimal solution of problem (5).

5. Implementation and Analysis

In this section, we use numerical simulations and compare our approach’s performance with others’ to evaluate the efficiency of the method. All the parameters in the simulations are different number of thick clients, number of brokers, traffic demand, network topology, and some big media files for the above method. The simulations are conducted in Java with JDK-7u7-i586 and Netbeans-7.2. We consider 2 cases: with

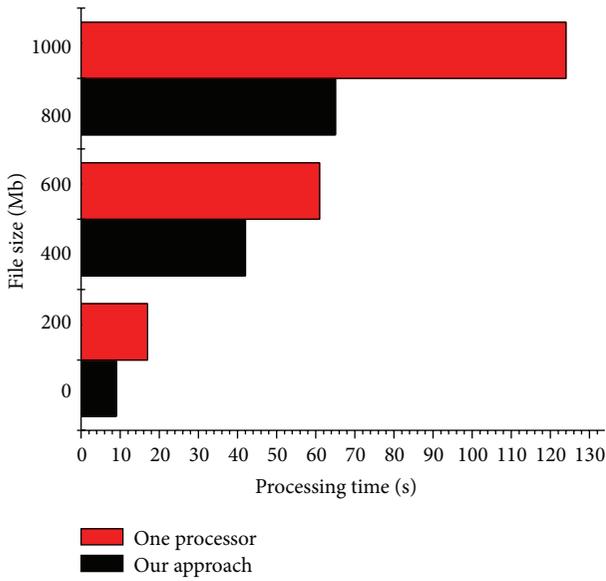


FIGURE 4: Comparison without user mobility.

and without user mobility. The implementation result in Figure 4 proves that, without user mobility, related to the processing time of transferring big media data from a cloud provider to a thin client, our approach leads to a better performance than the one which has one processor receiving data.

In addition, suppose that users move at the same speed; we examine the migration time of “state” data distribution from a current broker to a destination broker for thin clients. The results reflected in Figures 5 and 6 indicate that our method has worse objective value than Greedy, but better than Random method. In the meantime, the execution time of our approach is better than Greedy but worse than Random. After all, when we sum the time of data distribution from cloud provider to brokers and from brokers to thin clients, the total execution time of our proposal is still more efficiency.

6. Conclusion

In this paper, for efficient delivery of data, we present an architecture of thin-thick client collaboration. Our objective is to optimize data distribution between cloud customers and providers. With the user being mobile, expected QoS can be achieved. Furthermore, we conducted simulations to evaluate our method. Through the simulation results, it has been seen that our system performance is better than the existing ones. The future work includes advancing the research proposal into real-world implementation, firstly in testbed environments before expanding it into larger-scale production deployment. With the planned implementation, hopefully we can thoroughly observe the real-world operation and work out any rising problems or shortcomings should they be occurring. Moreover, we will focus more on improving the quality of service in order to bring better cloud service experience.

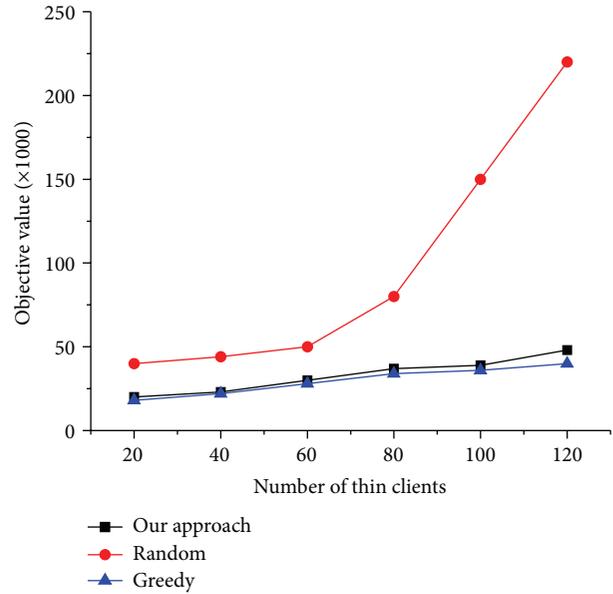


FIGURE 5: Objective value comparison with user mobility.

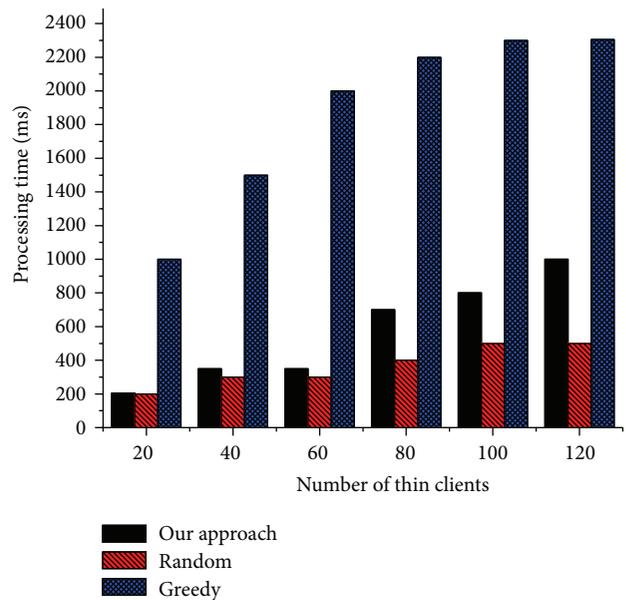


FIGURE 6: Execution time comparison with user mobility.

Conflict of Interests

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

Acknowledgment

This research was supported by the MSIP (Ministry of Science, ICT & 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).

References

- [1] Gartner Inc, *Gartner Says Worldwide Enterprise IT Spending to Reach #.7 Trillion in 2012*, 2013, <http://www.gartner.com/newsroom/id/1824919>.
- [2] Gartner Inc, *Gartner Says Worldwide PC, Tablet and Mobile Phone Combined Shipments to Reach 2.4 Billion Units in 2013*, 2013, <http://www.gartner.com/newsroom/id/2408515>.
- [3] X. Nan, Y. He, and L. Guan, "Optimal resource allocation for multimedia cloud based on queuing model," in *Proceedings of the 3rd IEEE International Workshop on Multimedia Signal Processing (MMSP '11)*, pp. 1–6, November 2011.
- [4] Y. Pan and J. Zhang, "Parallel programming on cloud computing platforms—challenges and solutions," *Journal of Convergence*, vol. 3, no. 4, pp. 23–28, 2012.
- [5] 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.
- [6] F. Sardis, G. Mapp, J. Loo, M. Aiash, and A. Vinel, "On the investigation of cloud-based mobile media environments with service-populating and QoS-aware mechanisms," *IEEE Transactions on Multimedia*, vol. 15, no. 4, pp. 769–777, 2013.
- [7] H. J. La and S. D. Kim, "A self-stabilizing process for mobile cloud computing," in *Proceedings of the IEEE 7th International Symposium on Service-Oriented System Engineering (SOSE '13)*, pp. 454–462, March 2013.
- [8] G. Huerta-Canepa and D. Lee, "A virtual cloud computing provider for mobile devices," in *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing and Services: Social Networks and Beyond (MCS '10)*, San Francisco, Calif, USA, June 2010.
- [9] T.-J. Kim, B.-G. Kim, C.-S. Park, and K.-S. Jang, "Efficient block mode determination algorithm using adaptive search direction information for scalable video coding (SVC)," *Journal of Convergence*, vol. 5, no. 1, pp. 14–19, 2014.
- [10] H. Cho and M. Choi, "Personal mobile album/diary application sevelopment," *Journal of Convergence*, vol. 5, no. 1, pp. 32–37, 2014.
- [11] U. Habiba, M. Imdadul Islam, and M. R. Amin, "Performance evaluation of the VoIP services of the cognitive radio system, based on DTMC," *Journal of Information Processing Systems*, vol. 10, no. 1, pp. 119–131, 2014.
- [12] H. Luo and M.-L. Shyu, "Quality of service provision in mobile multimedia—a survey," *Human-Centric Computing and Information Sciences*, vol. 1, p. 5, 2011.
- [13] S. Uppoor, M. D. Flouris, and A. Bilas, "Cloud-based synchronization of distributed file system hierarchies," in *Proceedings of the IEEE International Conference on Cluster Computing Workshops and Posters, Cluster Workshops*, pp. 1–4, September 2010.
- [14] E. Cuervoy, A. Balasubramanian, D. K. Cho et al., "MAUI: making smartphones last longer with code offload," in *Proceedings of the 8th Annual International Conference on Mobile Systems, Applications and Services (MobiSys '10)*, pp. 49–62, June 2010.
- [15] S. M. Ross, *Introduction to Probability Models*, Elsevier, University of Southern California, Los Angeles, Calif, USA, 10th edition, 2010.
- [16] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, 2010.
- [17] X. Luo, "From augmented reality to augmented computing: a look at cloud-mobile convergence," in *Proceedings of the International Symposium on Ubiquitous Virtual Reality (ISUVR '09)*, pp. 29–32, Gwangju, Republic of Korea, July 2009.
- [18] A. Bhattacharya, W. Wu, and Z. Yang, "Quality of experience evaluation of voice communication: an affect-based approach," *Human-centric Computing and Information Sciences*, vol. 2, p. 7, 2012.
- [19] J. Delgado, S. M. Sadjadi, L. Fong, Y. Liu, N. Bobroff, and S. Seelam, "Efficiency assessment of parallel workloads on virtualized resources," in *Proceedings of the 4th IEEE/ACM International Conference on Cloud and Utility Computing (UCC '11)*, pp. 89–96, December 2011.
- [20] V. Stantchev and C. Schrofer, "Negotiating and enforcing QoS and SLAs in grid and cloud computing," in *Proceedings of the International Conference in Grid and Pervasive Computing (GPC '09)*, 2009.
- [21] R. Hussain and H. Oh, "Cooperation-aware VANET clouds: providing secure cloud services to vehicular ad hoc networks," *Journal of Information Processing Systems*, vol. 10, no. 1, pp. 103–118, 2014.
- [22] B. Kim, C.-H. Youn, Y.-S. Park, Y. Lee, and W. Choi, "An adaptive workflow scheduling scheme based on an estimated data processing rate for next generation sequencing in cloud computing," *Journal of Information Processing Systems*, vol. 8, no. 4, pp. 555–566, 2012.
- [23] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *Proceedings of the ACM Conference on Data Communication (SIGCOMM '08)*, pp. 63–74, August 2008.

