

Improvement of QoS in Mobile Cloud Computing

Debabrata Sarddar¹ and Priyajit Sen²

¹Assistant Professor, Department of CSE, University of Kalyani, Kalyani, Nadia,
West Bengal, India

²M. Tech(Pursuing), Department of CSE, University of Kalyani, Kalyani, Nadia,
West Bengal, India

dsarddar1@gmail.com; priyajit91@gmail.com

Abstract

The concepts of cloud and service-oriented computing provide users with tremendous opportunities for arranging a large variety of services to accomplish desired tasks. At the same time, accessing service clouds for fast development in mobile devices has made it a typical instrument. However, the usage of mobile devices and accessing service clouds can be affected for the limited battery power. Power management is an important issue for mobile devices. We have proposed a mechanism for the enhancement of the quality of service in mobile cloud computing to predict performance of a set of cloud service modes before selecting a service mode and to select the best service mode based on previous prediction results.

Keywords: *cloud computing, mobile cloud computing, concept of mobile cloud, Quality of service, modeling, service oriented programming*

1. Introduction

Cloud Computing is established as the newest big shift in the computing. For some, the notion of managing and storing data in a centralized, ‘Always-On’ repository that can be accessed from anywhere is the key concept. Often many Service Providers speak about different mobile strategies, and how each of these various services offers a tuned, modified or enhanced experience for a mobile.

1.1 Concept of Mobile Cloud Computing

Mobile Cloud computing is related concept of cloud computing and it brought the services like on demand access, no on-premise software. Mobile Cloud Computing, use network capabilities alone to deliver the desired service to customers, as well as charge them for providing service. Mobile Cloud Computing can permit you to reserve network bandwidth, thus it confirms timely delivery of information to a user.

1.2 Mobile Cloud Computing: Change is Required

Mobile world is dependent on two factors. One is Network Stability and the other is Handset availability. Since mobile phones do not have adequate processing power or memory to support huge amount of data, cloud computing seems to be the ideal solution for these mobile phone users. Cloud computing will allow these mobile phone users to have the same amount of data access like “smart” phone users have stored their data in the phone. This extra advantage of Cloud Computing allows developers and mobile companies to start targeting a larger market than only “smart” phone users, which in turn will give Cloud Computing more thrust in near future.

1.3 The Need of Mobile Cloud Computing:

The case for mobile cloud computing can be claimed by considering the exclusive advantages of empowered mobile computing, and a wide range of potential mobile cloud applications have been recognized in the literature. These applications fall into many areas like natural language processing, image processing, sharing Internet access, sensor data applications, sharing GPS, querying crowd computing and multimedia search. However, as described in, applications that involve distributed computation do have certain common characteristics, like consuming data with easily visible segment boundaries, and the time to recombine partial results into a complete result must also be small. An example is string matching/manipulation like grep and word frequency counters. The different scenarios and applications presented in recent literature are described in detail below:

1.3.1 Image Processing:

In this paper [1], the authors have experimented with running GOCR, an optical character recognition (OCR) program on a pool of mobile devices. In an actual life scenario, this would be beneficial in a case of a foreign traveler who takes an image of a street sign, performs OCR to extract the words then translates words into a known language. A similar scenario is given in paper [2] where a foreign tourist Peter is visiting a museum in South Korea. He sees an interesting exhibit, but cannot understand the description since it is in Korean. He takes a picture of the text, and starts an OCR app on his phone. Unfortunately his phone lacks the resources to process the text. Although he could connect to a remote server via the Internet, that would mean he use roaming data which is too expensive. Instead, his device scans for nearby users/devices who are also interested in reading the description, and requests sharing their mobile resources for the task collaboratively. Those who are concerned in this common processing task create an ad hoc network with Peter and together, their mobile cloud is able to extract the text, and then translate it to English.

1.3.2 Natural Language Processing:

As mentioned above, language translation is one of the likely applications, and this is mentioned in paper [3] as a useful tool for foreign travelers to communicate with locals. Translation is a feasible candidate since different sentences and paragraphs can be translated independently, and this is experimentally explored in [4] using Pangloss-Lite. Text-to-speech is also mentioned in paper [5], where a mobile user may desire having a file read to them, especially in case of the visually impaired.

1.3.3 Crowd Computing:

Video recordings from multiple mobile devices can be spliced to construct a single video that covers the entire event from different angles, and perspectives [6]. In paper [7], two scenarios of this nature are described in detail: 'Lost child' and 'Disaster relief'.

1.3.4 Sharing GPS/Internet Data:

It is more effective to share data among a group of mobile devices that are near each other, through local-area or peer-to-peer networks. This is not only cheaper, but faster also [8]. Rodriguez et al. [9] present a case study of a hiking party at Padjelanta National Park, which is a deserted land in the Arctic Circle lacking power access points and network coverage. A data set contains Bluetooth scans for discovering devices and GPS reads of 17 persons.

1.3.5 Sensor Data Applications:

Now a day's most of the mobile phones comes with sensor and reads from sensors such as GPS, light sensor, accelerometer, clock, thermometer, microphone and compass can be time stamped and linked with other phone readings. Execution of queries is done on such data for gather valuable information. These queries could be “what is the average temperature of nodes within a mile of my location?” or “what is the distribution of velocities of all nodes within half a mile of the next highway on my current route?” Sample applications for this are traffic reporting, sensor maps, and network availability monitoring [10].

1.3.6 Mobile Commerce Applications:

Mobile commerce (m-commerce) is like a business model for commerce by mobile devices.). Mobile commerce applications are used for many mobile transactions and payments, and mobile ticketing and mobile messaging. The most common problem of m-commerce applications are low network bandwidth, high complexity of mobile device configurations, and security. Therefore, m-commerce applications are combined into cloud computing environment to report these issues.

1.3.7 Mobile Learning

Mobile learning (m-learning) is considered based on electronic learning (e-learning) and mobility. Cloud-based m-learning applications solve the limitations of traditional m-learning applications e.g. high cost of devices and network, low network transmission rate, and limited educational resources [11], [12], [13].

1.3.8 Mobile Gaming

Mobile game can completely offload game engine requiring large computing resource to the server in the cloud, and gamers only interact with the screen interface on their devices. It is found that instead of offloading all codes to the cloud for processing, MAUI partitions the application codes at a runtime based on the costs of network communication and CPU on the mobile device to maximize energy savings given network connectivity. The objective is to maximize the user experience given the communications and computing costs.

1.3.9 Other Practical Applications:

The cloud computing becomes a useful tool to help mobile users share photos and video clips efficiently and tag their friends in popular social networks as Twitter and Facebook. A cloud becomes the most effective tool when mobile users require searching services (*e.g.*, searching information, location, images, voices, or video clips). Examples of such applications are discussed below,

A. Multimedia Search:

Mobile devices store several types of multimedia content such as videos, photos, and music. For example, Shazam is a music identification service for mobile phones that searches for similar songs in a central database. In the context of the mobile cloud, the searching could be executed on the contents of nearby phones [14].

B. Social Networking:

Since sharing user content is a popular way by which we interact with friends on social networks such as Facebook, integrating a mobile cloud into social networking

infrastructure. It can perform automatic sharing, p2p multimedia access and reduce the need to back up and serve all of these data on huge servers.

2. Related Works

There are several approaches to statically or dynamically offload applications at runtime to reduce computation burden on mobile devices under mobile cloud computing environment. Chun's work presents an approach to dynamic partitioning of applications between resource-constrained mobile devices and clouds, to adapt workloads dynamically [15]. They present main issues for dynamic partitioning such as how to dynamically execute partitioned functionalities and how to determine the right configuration at runtime. They elaborate the system, *Clone Cloud*, which enables unmodified mobile applications running on virtual machine to seamlessly offload computation to clouds [16]. The system consists of three components; *static analyzer* identifying legal choices for migration and re-integration points in the code, *dynamic profiler* creating a profile tree which is a representation of execution on a single platform with execution time and energy consumption, and *optimization solver* determining the most appropriate migration method by using the profile tree. This work only considers offloading pre-determined components to the designated cloud node and present less detailed methods on performed by each component. Satyanarayanan's work presents an approach to execute the computation intensive software from mobile device by using virtual machine (VM) technologies [17]. The computation is migrated to *cloudlet*, which is a trusted, resource-rich computing computer or a cluster of computers nearby a user, rather than remote cloud. The migration is realized by VM synthesis where small VMs delivered from mobile devices is overlaid to cloud let having the base VM. This work only conceptually describes how the MV synthesis performs without mentioning the overall architecture. Cuervo's work presents a system that enables fine grained energy-aware offload of mobile code to the infrastructure, called MAUI [18]. MAUI adopts client-server architecture style, and both sides embed three components; proxy, profiler, and solver. Profiler determines whether the method invocation runs locally or remotely by using device, program, and network profiling. And, solver finds the optimal program partition strategy in terms of energy consumption and latency. By using reflection, server proxy extracts and executes remote methods which are pre-defined at design time. With MAUI, a code is offloaded to the designated server, which is quite constrainedly applied to MCC. Malek's work presents an architecture-driven framework supporting the entire life-cycle of a mobile software system [19]. It consists of tools for assessing, deploying, and migrating mobile software systems at runtime. This work focuses on elaborating the functionality of the tools without detailed realization mechanisms. Han's work presents an adaptive software architecture supporting component migration and redeployment at runtime [20]. The work proposes four types of connectors for adaptive architecture; *link*, *pull*, *copy*, and *stamp*. However, implementation issues of the connectors are not treated. In addition, there has been research on proposing middleware or framework for supporting offloading mechanisms such as Li's work [21], Yang's work [22], Ye's work [23], and Hung's work [24].

3. Proposed Work

3.1 System Architecture

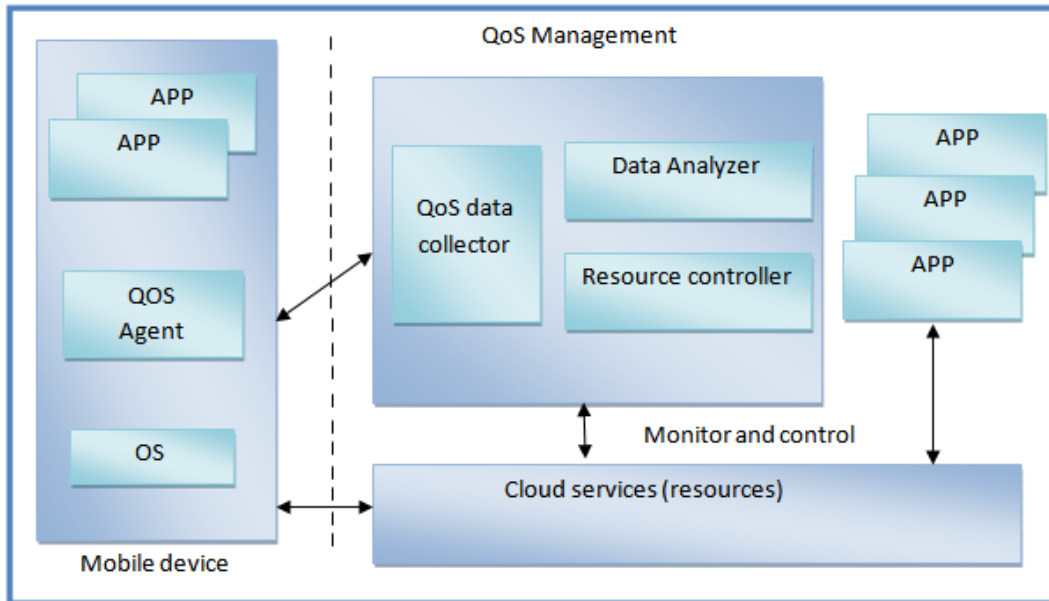


Figure 1. QoS Framework for Mobile Cloud Computing

Figure 1 shows a QoS management framework for mobile cloud computing. In a mobile device, a QoS agent monitors QoS status at run time, e.g., percentage of memory and CPU consumption, connection speed, remaining battery percentage and packet loss rate, etc. The QoS status will be reported to a QoS management center in cloud side. The QoS management center aggregates and analyzes the huge set of QoS data, and dynamically adjusts resources to meet QoS requirements of each mobile cloud service. Based on the QoS management framework, we apply several modes of mobile cloud services. Each mode contains multiple services, mechanisms and resource configuration schemes. A cloud service mode is a specific configuration to guarantee the QoS requirements for a cloud service. Notably, the mobile cloud computing platform can provide multiple similarly functioned services that can satisfy the demand of an integrated service. Especially, the QoS requirements of a service can be assured by selecting suitable service modes. We propose a self adaptive QoS management process for mobile cloud services. In this process, QoS Predication is a mechanism to predict performance of a set of cloud service modes before selecting a service mode. Mode selection is a mechanism to select the best service mode based on previous prediction results. QoS Assessment is a mechanism to monitor and assess the QoS status according to users' QoS requirements. For the QoS requirements of a service, the QoS values can be predicted by assuming a service mode is selected. Based on prediction results, a service mode can be selected and set as system configuration. The QoS assessment mechanism evaluates the QoS status by monitoring the performance of the cloud service. According to the assessment results, the system adjusts the parameters of QoS control model to reflect real status. The adjustment happens when the evaluation result is below a threshold that is defined by users. The process runs over to achieve the self adaptive QoS management in the dynamic mobile cloud environment. In particular, the QoS management supports context awareness by adaptively selecting a proper set of service modes that can always ensure the quality of cloud services.

3.2 Modeling

We apply Fuzzy Cognitive Map to model the factors considered in adaptive QoS management.

FCM specifies the interconnections and influences between nodes. It also permits updating the construction of the graph, such as adding or deleting an interconnection or a node [25]. FCM is

a useful method in modeling and control of complex systems. It helps the system designer in decision analysis and strategic planning. Based on the FCM theory, a stable control performance could be anticipated based on a specific FCM configuration. Thus, we can utilize it to predict the performance of cloud service modes in order to select the best one.

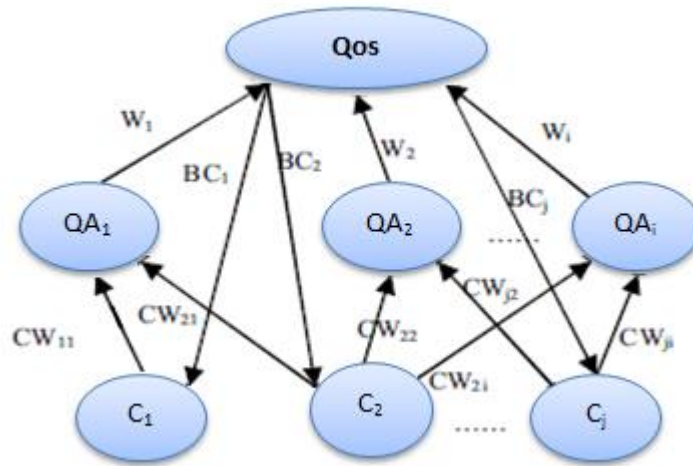


Figure 2. QoS Management Modeling based on FCM

We propose a QoS management model with FCM as illustrated in Figure 2. The model includes three layers of nodes. The top layer node represents QoS values of a cloud service or an integrated set of cloud services. The middle layer includes the QoS parameters QA_i ($i = 1, \dots, n$) of the service. The bottom layer includes cloud service modes C_j ($j = 1, \dots, m$). Each cloud service mode has impact on QoS parameters and therefore on QoS value. On the other hand, the QoS value also has impact on the effectiveness of each service mode. Here, we assume that each service mode is independent from each other. The value of each node is influenced by the values of its connected nodes with appropriate weights and its previous value. Thereby, we apply an addition operation to account for both. The QoS value can be described as:

$$Q = f\left(\sum_{i=1}^n w_i V_{QA_i} + Q^{old}\right) \text{-----} \quad (1)$$

Such that

$$\sum_{i=1}^n w_i = 1$$

where w_i is the weight indicating the importance rate of the QoS attribute QA_i regarding how much this attribute is considered at the QoS assessment. The w_i can be set based on the user's criteria (in practice can be selected from a profile). V_{QA_i} is the value of the QoS attribute and Q^{old} is the respective values of in the previous iteration. V_{QA_i} can be calculated according to the following formula:

$$V_{QA_i} = f\left(\sum_{j=1}^m cw_{ji} V_{C_j} B_{C_j} + V_{QA_i}^{old}\right) \text{-----} \quad (2)$$

where cw_{ji} is the influence factor of service mode C_j on QA_i , cw_{ji} is set based on the impact of C_j on QA_i . A positive cw_{ji} indicates a positive influence of C_j on QA_i . A negative cw_{ji} implies a negative influence of C_j on QA_i . BC_j is the selection factor of C_j , which can be either 1 if C_j is applied or 0 if C_j is not applied. Notably, BC_j indicates the current cloud computing platform configuration regarding which service mode is applied for a cloud service. The value of the service mode can be calculated using

$$V_{C_j} = f(Q \cdot B_{C_j} + V_{C_j}^{old}) \quad \text{----- (3)}$$

We apply the Sigmoid function as a threshold function: $f(x) = 1/(1 + e^{-\alpha x})$, e.g., $\alpha = 2$ to map node values $V_{QA_i} V_{C_j} Q$, into $[0, 1]$. Note that $V_{QA_i} V_{C_j} Q \in [0, 1]$, $w_i \in [0, 1]$, and $[-1, 1]$ $cw_{ji} \cdot V_{QA_i}^{old}$ and $V_{C_j}^{old}$ are the respective values of V_{QA_i} , and V_{C_j} in the previous iteration.

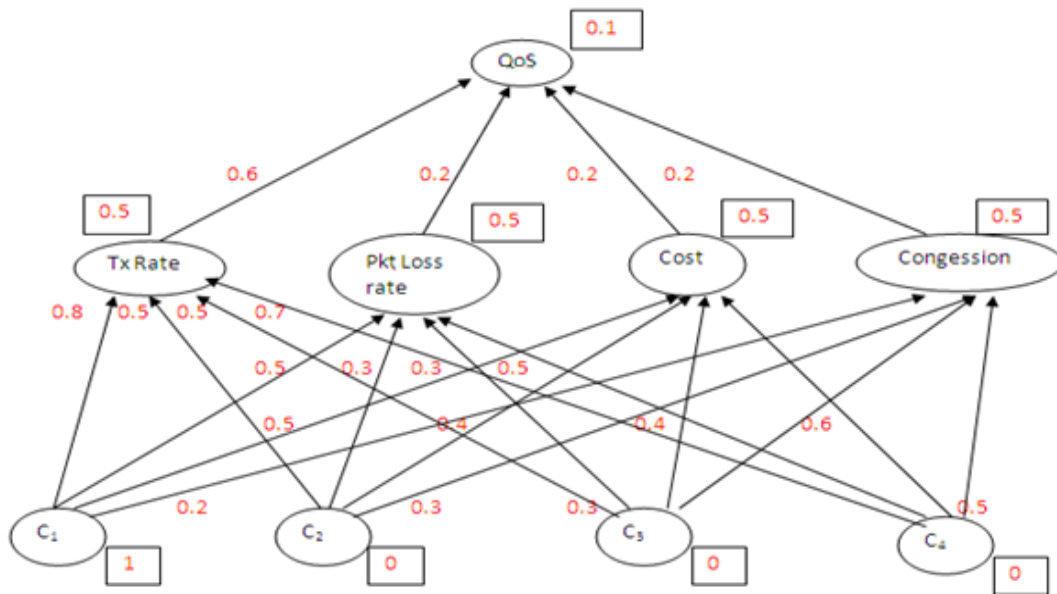


Figure 3. Simulation Configurations

4. Simulation

The simulation is based on an example as shown in Figure 2. It is a multi-to-multi video conference service. The QoS properties include three variables, i.e., 1 QA transmission rate, 2 QA packet loss, and 3 QA cost. There are three service modes offered by the cloud computing platform:

C_1 : High configuration mode with high cost

C_2 : Medium configuration mode with medium cost

C_3 : Low configuration mode with low cost

$w_1 = 0.6$, $w_2 = 0.2$, and $w_3 = 0.2$ $w_4 = 0.2$ Note that four service modes (C_1 , C_2 , C_3 and C_4) and their influence factors are specified in the system's profile, an example used in our simulation is shown in Figure 3 First, we run the simulation by giving different initial QoS values. After a few times iteration, the QoS value calculation becomes stable no matter which initial value. This shows that the QoS prediction algorithm is robust irrespective of initial QoS value. Similarly, we run the simulation by changing the initial value of QA_i .

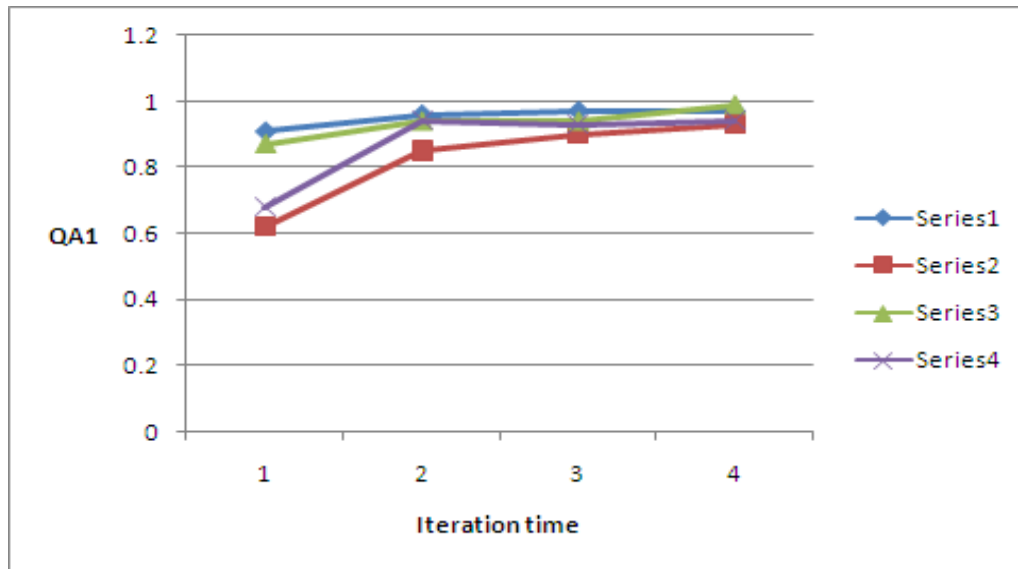


Figure 4. QoS Value Calculation with Different Initial QoS Value

5. Conclusions and Future Work

In this paper, we have proposed an adaptive QoS management system for mobile cloud computing. This solution facilitates QoS prediction, establishment, assessment and assurance. We have introduced the influence of QoS properties and service modes into the model, which supports adaptive QoS management according to QoS assessment based on runtime QoS observation. We applied the FCM theory into QoS management and showed its practical effectiveness through case study. We have reported the simulation-based experimental results to verify the proposed system and demonstrated its effectiveness and benefits. We have contributed a practical solution that can react against QoS unsatisfactory adaptively at service runtime and handle the QoS requests with different criteria.

References

- [1] J. Cheng, R.K. Balan, M. Satyanarayanan, Exploiting rich mobile environments, Technical Report, 2005.
- [2] G. Huerta-Canepa, D. Lee, A virtual cloud computing provider for mobile devices, in: Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond, MCS'10, ACM, New York, NY, USA, 2010, pp. 6:1–6:5.
- [3] R.E. Frederking, R.D. Brown, The pangloss-lite machine translation system, in: Proceedings of the Second Conference of the Association for Machine Translation in the Americas, pp. 268–272.
- [4] E.E. Marinelli, Hyrax: cloud computing on mobile devices using MapReduce, Masters Thesis, Carnegie Mellon University, 2009.
- [5] M. Satyanarayanan, Mobile computing: the next decade, in: Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & #38; Services: Social Networks and Beyond, MCS'10, ACM, New York, NY, USA, 2010, pp. 5:1–5:6.
- [6] N. Vallina-Rodriguez, J. Crowcroft, Erdos: achieving energy savings in mobile OS, in: Proceedings of the Sixth International Workshop.
- [7] X. Yang, T. Pan, and J. Shen, “On 3G Mobile E-commerce Platform Based on Cloud Computing,” in Proceedings of the 3rd IEEE International Conference on Ubi-Media Computing (UMedia), pp. 198 - 201, August 2010.
- [8] J. Dai, and Q. Zhou, “A PKI-based mechanism for secure and efficient access to outsourced data,” in Proceedings of the 2nd International Conference on Networking and Digital Society (ICNDS), vol. 1, pp. 640, June 2010.
- [9] Z. Leina, P. Tiejun, and Y. Guoqing, “Research of Mobile Security Solution for Fourth Party Logistics,” in Proceedings of the 6th International Conference on Semantics Knowledge and Grid (SKG), pp. 383 - 386, January 2011 .

- [10] X. Chen, J. Liu*, J. Han, and H. Xu, "Primary Exploration of Mobile Learning Mode under a Cloud Computing Environment," in Proceedings of the International Conference on E- Health Networking, Digital Ecosystems and Technologies (EDT), vol. 2, pp. 484 -487, June 2010.
- [11] H. Gao and Y. Zhai, "System Design of Cloud Computing Based on Mobile Learning," in Proceedings of the 3rd International Symposium on Knowledge Acquisition and Modeling (KAM), pp. 293 - 242, November 2010.
- [12] Jian Li, "Study on the Development of Mobile Learning Promoted by Cloud Computing," in Proceedings of the 2nd International Conference on Information Engineering and Computer Science (ICIECS), pp. 1, December 2010.
- [13] I. Foster and C. Kesselman, The Grid: Blueprint for a New Computing Infrastructure: Morgan Kaufmann, 1998.
- [14] J. Napper and P. Bientinesi, "Can cloud computing reach the top500?," in Combined Workshops on UnConventional High Performance Computing Workshop plus Momory Access Workshop, 2009, pp. 17-20.
- [15] Y. Chen, V. Paxson, and R. Katz, "What's New About Cloud Computing Security?," 2010.
- [16] A. Leinwand, "The Hidden Cost of the Cloud: Bandwidth Charges," <http://gigaom.com/2009/07/17/the-hidden-cost-of-the-cloud-bandwidthcharges/>, 2009.
- [17] J. Gray, "Distributed computing economics," ACM Queue, vol. 6, pp. 63-68, 2008.
- [18] M. May, "Forecast calls for clouds over biological computing," Nature Medicine, vol. 16, p. 6, 2010.
- [19] M. Nelson, "Building an Open Cloud," Science, vol. 324, p. 1656, 2009.
- [20] B. Sotomayor, R. Montero, I. Llorente, and I. Foster, "Virtual Infrastructure Management in Private and Hybrid Clouds," IEEE Internet Computing, vol. 13, pp. 14-22, 2009.
- [21] T. Harmer, P. Wright, C. Cunningham, and R. Perrott, "Provider- Independent Use of the Cloud," in The 15th International European Conference on Parallel and Distributed Computing, 2009, p. 465.
- [22] "Unified Cloud Interface Project," <http://code.google.com/p/unifiedcloud/>.
- [23] D. Nurmi, R. Wolski, C. Grzegorzczak, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "The eucalyptus open-source cloudcomputing system," presented at the Proceedings of Cloud Computing and Its Applications, 2008.
- [24] "Sun Microsystems Unveils Open Cloud Platform", <http://www.sun.com/aboutsun/pr/2009-03/sunflash.20090318.2.xml>, 2009.
- [25] M. Zeller, R. Grossman, C. Lingenfelder, M. Berthold, E. Marcade, R. Pechter, M. Hoskins, and R. Holada, "Open standards and cloud computing: KDD-2009 panel report," in KDD, Paris, France, 2009, pp. 11-1

Authors



Debabrata Sarddar, Assistant Professor in the Department of Computer Science and Engineering, University of Kalyani, Kalyani, Nadia, West Bengal, India. He has done PhD from Jadavpur University. He has completed his M.Tech in Computer Science & Engineering from DAVV, Indore in 2006, and his B.E in Computer Science & Engineering from NIT, Durgapur in 2001. He has published around 200 research papers in different journals and conferences. He has written two books and also published two book chapters. His research interest includes Mobile Computing, Wireless Sensor Network and Cloud Computing.



Priyajit Sen is presently pursuing M.Tech in Computer Science and Engineering at the Department of Computer Science and Engineering, University of Kalyani, Kalyani, Nadia, West Bengal, India. He has completed his MCA from Department of Computer Science and Engineering, University of Kalyani, Kalyani, Nadia, West Bengal, India in 2015. He has published around 6 research papers in different journals and conferences. His research interest includes Mobile Computing, Wireless Sensor Network, IOT and Cloud Computing.

