International Journal of Advanced Computer Technology (IJACT)

ISSN:2319-7900

MODELING AND ANALYSIS OF COMPLETION TIME AND ENERGY CONSUMPTION OF APPLICATIONS IN MOBILE CLOUD COMPUTING ENVIRONMENTS

Young-Chul Shim, Hongik University

Abstract

We consider the execution of an interactive application in a mobile cloud computing environment including a local cloudlet and a remote cloud. We introduce models for computing completion time and energy consumption of an interactive application when its offloadable portion is executed on a mobile device, a local cloudlet, or a remote cloud. Applying practical numbers to some of the parameters in the models, we derive conditions under which computation offloading to either a local cloudlet or a remote cloud becomes beneficial in terms of completion time and energy consumption.

Introduction

Since the emergence of the concept of cloud computing, it is getting more widely adopted and deployed in the IT industry sector and receiving more attention from computer scientists and engineers. NIST defines cloud computing to be a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [1]. Examples of well-known cloud computing systems include Amazon AWS, Microsoft Azure, Google AppEngine, and Rackspace CloudServers [2], [3].

Mobile devices such as smartphones and tablet PCs are becoming more and more essential part of our life. People use mobile devices to not only communicate with others but also run application programs and store information gathered from their daily activities. But mobile devices suffer from the critical drawback, lack of resources. They have limited computation and storage capability and, more seriously, are very much restricted in their battery power.

To continue enjoying the convenience of mobile devices while making up for their weaknesses, people introduced the concept of mobile cloud computing [4]-[6]. In mobile cloud computing environments, computation and/or storage requirements on mobile devices can be offloaded to outside cloud computing environments and mobile users can finish their programs faster, store more information, and save the battery power of their mobile devices.

A large fraction of applications that mobile users run in the mobile cloud computing environment will be interactive. People play a chess game with their mobile devices but finding the optimal next movement requires tremendous amount of computation and, therefore, has been usually calculated on very powerful machines. In an application called content-based image retrieval, people may want to retrieve photos containing a certain image from a large file of photo collections, which will require a large amount of image matching computation. People may design a product such as a house or a machine with their mobile devices and performing this mobile computer-aided design activity usually requires solving many complex partial differential equations and demands extremely fast floating point computation. People can also benefit from the technology of augmented reality with their mobile devices. When they go to a store to buy a certain product such as furniture, they will want to check whether the chosen furniture goes well with their house. They can get the information about the chosen furniture from the store computer, send it to the site which stores information about their house, request to compose these two data, and view the results from many different aspects to make the buying decision.

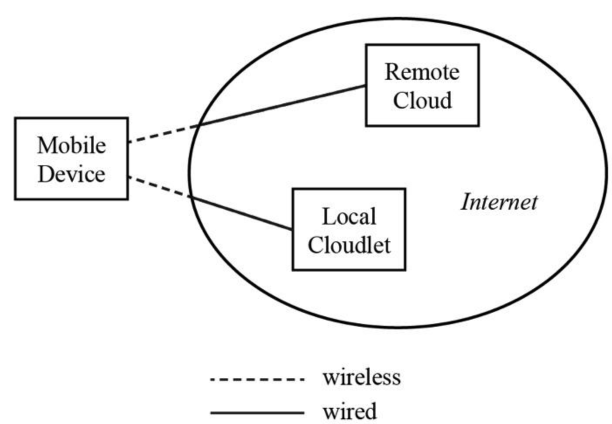

Figure 1. A Mobile Cloud Computing Environment

Cloud computing systems may be located far away from a mobile user and the communication latency between them can become non-negligible. If running an application requires a large number of interactions, the non-negligible latency may produce a negative effect on the response time and battery power saving. Recently the concept of cloudlets is intro-duced. Cloudlets are decentralized and widely-dispersed Internet infrastructure whose compute cycles and storage resources can be leveraged by nearby mobile users [7]. As server machines are becoming cheaper, we can make each node of a cloudlet as powerful as that of a cloud center and deploy a large number of cloudlets. With this we envision a mobile cloud computing environment as Figure 1 in which a mobile user can choose the location for running his program among a mobile device, a local cloudlet, or a remote cloud.

Some researchers derived conditions which must be satisfied to make computation offloading profitable to a mobile user [8], [9]. But their models about the computation and data transfer requirements and power consumption on mobile devices are too simplistic. And they do not consider propagation delay and, therefore, do not make distinction between local cloudlets and remote clouds. Some researchers developed static or dynamic program partitioning methods which help determine which portion of a program is to be offloaded to reduce response time and energy consumption [10], [11]. But their results apply to only specific programs in specific environments and do not provide general guidelines for when a portion of program should be offloaded to which outside environments, local cloudlets or remote clouds, and how much benefit can be obtained from the computation offloading.

In this paper we consider interactive applications running in a mobile cloud computing environment including both local cloudlets and remote clouds. We provide models for calculating completion time and energy consumption of an interactive application. In the model we consider factors including computation and data requirements of an application, processing speed, bandwidth, propagation delay, and energy usage of a mobile device in three states: computation, data transfer, and idle. The application can be run completely on a mobile device or the offloadable portion can be executed on either a local cloudlet or a remote cloud. Using the proposed model and applying typical values to the parameters in the model, we derive conditions that must be satisfied if executing the offloadable portion on either a local cloudlet or a remote cloud is to be beneficial.

The rest of the paper is organized as follows. Section 2 explains the execution model of an interactive application in a mobile cloud computing environment. The models for computing completion time and energy consumption of a program in various environments are provided in Section 3. Section 4 describes the conditions that must be satisfied if computation offloading is to be beneficial and is followed by the conclusion in Section 5.

Executing an Interactive Application in a Mobile Cloud Computing Environment

Interactive applications that will be executed in a mobile cloud computing environment may have different characteristics. An application such as a chess game does not need any initial data representing the initial state to be loaded to invoke the program. But an application such as the content-based image retrieval requires initial data which is usually a large size file of photo collection, on which various kinds of image matching operations will be executed. A mobile CAD program can lie in between. If a completely new design is to be started, there is no initial design data be loaded but if an unfinished design is to be resumed, the design data that have been accumulated from the previous design activities should be loaded, Even for the applications for which initial data should be loaded , the location of initial data can vary. They can be on the mobile device with which a user will run the interactive application or they can be located on a remote cloud. Sometimes when data are collected and stored at a mobile node, they can be synchronously copied to a remote cloud server to prevent data loss. In this paper, we assume that the initial data are located in the mobile node, if any.

After the initial data loading phase, which is performed only when required, comes the interactive computation phase. In the interactive computation phase, an activity consisting of three steps are repeated until the user terminates the program. The three steps are input, process, and output steps. In the input step, input data are obtained from the input devices such as a microphone, a camera, a keypad, etc and can then be preprocessed. The preprocessing can consist of various kinds of activities and one example is data compression which is performed to reduce the amount of data before being sent to a remote node. Because the input step requires the use of input device of a mobile node, it should be executed on the mobile node. In the second step, the process step, the input data is processed to produce the requested result. Examples of this step include finding the optimal movement in a chess game, retrieving photos matching a certain image from a large file of photo collection, or performing a requested design action which may require a huge amount of complex calculation such as solving many partial differential equations. This second step tends to be very compute-demanding and is a very good candidate for being

ISSN:2319-7900

migrated to and executed on a remote fast node. The last step, the output step, receives the result from the process step and presents it to the user using the output devices on the mobile node. Therefore, this step should be performed on the mobile node. If the process step is executed on a remote fast node, a single iteration of the three step computation phase will proceed as in Figure 2.

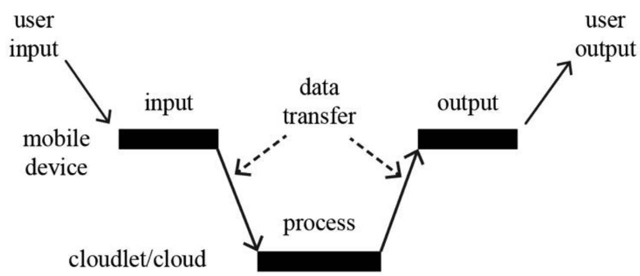

Figure 2. Remote Execution of a Process Step

Modeling Completion Time and Energy Consumption

If an interactive application requires C instructions for computation, these instructions can be divided into two parts: one that must be executed on a mobile node and the other that can be offloaded to a remote node, possibly a very fast cloud node. With the execution model described in the pre-vious section, the input and output steps belong to the first part and the process step belongs to the second part. In this paper we consider two candidates for the outside computa-tion node: a local cloudlet and a remote cloud. A local cloudlet may have less computation capacity than a remote cloud but is definitely closer from the mobile node and, therefore, incurs much smaller propagation delay when ex-changing data with a mobile node.

We consider two kinds of interactive applications: one which does not require initial data and the other which requires initial data. For those applications which require initial data, such as a large size file of photo collection in the content-based image retrieval application, we assume this initial data is stored in the mobile node. For these two kinds of applications, we calculate the completion time and energy consumption requirements and compare the results.

Now we will define symbols that will be used in this paper.

- N: Represents the number how many times the input-process-output stage is repeated during the interactive computation phase.
- C_M: Number of instructions that should be executed on a mobile node
- C_C: Number of instructions that can be offloaded to an outside node such as a local cloudlet or a remote cloud. These instructions are called offloadable instructions in this paper and they correspond to the process step in Figure 2. If there are N iterations in the computation phase, each process step executes CC/N = CC' instructions on average.
- S_M: Speed of a mobile node in terms of instructions/second.
- S_C: Speed of a cloud or cloudlet in terms of instructions/second. If we have to distinguish a local cloudlet from a remote cloud, we use SCL for the speed of a local cloudlet and SCR for a remote cloud.
- D_I: The size of the initial data needed for the operation of an interactive application.
- D_C: The amount of data that a mobile node exchanges with an outside node during the interactive computation phase. They are the data moved to and from the process step in Figure 2. If there are N iterations in the computation phase, each process step exchanges data of the amount DC/N = DC'.
- B_M: The bandwidth between a mobile and an outside node.
- T_P: The propagation delay between a mobile node and an outside node. This delay includes not only the delay proportional to the distance between two communicating nodes but also delays at the intermediate communication devices such as switches and routers. If we have to distinguish a local cloudlet from a remote cloud, we use TPL for a local cloudlet and TPR for a remote cloud.
- P_C: The energy consumed during computation by a mobile node in watts.
- P_I: The energy consumed during idle time by a mobile node in watts.
- P_T: The energy consumed to send and receive data by a mobile node in watts. Usually transmitting power is higher that receiving power, but we assume that they are the same in this paper for simplicity.
- M_N: Mobile node
- L_C: Local cloudlet
- R_C: Remote cloud

Now we will present models for completion time and energy consumption of an interactive application for the following two cases: one without any initial data and the other with initial data residing at a mobile node.

A. No Initial Data

ISSN:2319-7900

When all the computation is performed on a mobile node, all the (C_M+C_C) instructions are executed with the speed of S_M. There is no data exchange. So, the completion time T and the consumed energy E are

$$T(MN) = (C_M+C_C)/S_M \tag{1}$$
$$E(MN) = P_C \times (C_M+C_C)/S_M \tag{2}$$

When all the C_C instructions are executed on either a local cloudlet or a remote cloud, in addition to the time spent to execute instructions, extra time is needed to exchange the data of size D_C. The data exchange time consists of a data transmission time, which is defined to be *(data size)/bandwidth*, and the data propagation time between two nodes. Each iteration of an input-process-output step consists of two data transmissions, one for input and the other for output, and we assume that this three step operation is repeated N times. Thus the whole interactive computation phase involves 2N data transmissions and requires $2N \times T_P$ seconds for the whole propagation time. Therefore, the completion time and the consumed energy at the mobile node with C_C instructions offloaded to either a local cloudlet of a remote cloud become

$$T(LC \text{ or } RC) = C_M/S_M+C_C/S_C+2N \times T_P+D_C/B_M \tag{3}$$
$$E(LC \text{ or } RC) = P_C \times (C_M/S_M)+P_I \times (C_C/S_C+2N \times T_P) + P_T \times (D_C/B_M) \tag{4}$$

Note that in Equation (4) when instructions are executed on a local cloudlet or a remote cloud and exchanged data are moved to and from the mobile node (this means that data are in the medium), the mobile node stays in the idle mode. For Equations (3) and (4), S_C and T_P become S_{CL} and T_{PL} if a local cloudlet is used and become S_{CR} and T_{PR} if a remote cloud is used.

To compare the completion time and consumed energy, we computer $T(MN)$ – $T(LC$ or $RC)$ and $E(MN)$ – $E(LC$ or $RC)$ as follows

$$T(MN)-T(LC \text{ or } RC) = C_C(1/S_M-1/S_C)-2N \times T_P-D_C/B_M \tag{5}$$
$$E(MN)-E(LC \text{ or } RC) = C_C \times (P_C/S_M-P_I/S_C)- P_I \times (2N \times T_P)-P_T \times (D_C/B_M) \tag{6}$$

Now we consider the case in which CC instructions are of-floaded to either a local cloudlet or a remote cloud. In order to compare theses two cases we compute $T(RC)$ - $T(LC)$ and $E(RC)$ – $E(LC)$ as follows

$$T(RC)-T(LC) = C_C \times (1/S_{CR}-1/S_{CL})+2N \times (T_{PR}-T_{PL}) \tag{7}$$
$$E(RC)-E(LC) = P_I \times \{C_C\,(1/S_{CR}-1/S_{CL})+2N \times (T_{PR}-T_{PL})\} \tag{8}$$

We see that Equation (8) is the same as Equation (7) except that Equation (8) is P_I times of Equation (7).

B. Initial Data at a Mobile Node

In this subsection we consider the case in which initial data is required for the application to operate and the data reside at a mobile node.

If all the computation is performed at a mobile node without any computation offloaded to an outside node, the completion time and the consumed energy at the mobile node are the same as Equations (1) and (2), respectively.

If C_C instructions are to be executed at an outside node, the initial data D_I should be transferred to that outside node. This incurs more time $2 \times T_P+D_I/B_M$ and more energy $2 \times P_I \times T_P+P_T \times (D_I/B_M)$. For simplicity we assume $N >> 2$ and we can ignore the term due to $2 \times T_P$. Then, the completion time and the consumed energy at the mobile node with C_C instructions offloaded to either a local cloudlet or a remote cloud become

$$T(LC \text{ or } RC) = C_M/S_M+C_C/S_C+2N \times T_P+(D_I+D_C)/B_M \tag{9}$$
$$E(LC \text{ or } RC) = P_C \times (C_M/S_M)+P_I \times (C_C/S_C+2N \times T_P) + P_T \times ((D_I+D_C)/B_M)) \tag{10}$$

To compare the completion time and consumed energy, we compute $T(MN)$ – $T(LC$ or $RC)$ and $E(MN)$ – $E(LC$ *or* $RC)$ as follows.

$$T(MN)-T(LC \text{ or } RC) = C_C \times (1/S_M-1/S_C)- 2N \times T_P-(D_I +D_C)/B_M \tag{11}$$
$$E(MN)-E(LC \text{ or } RC) = C_C \times (P_C/S_M-P_I/S_C)-P_I \times (2N \times T_P)-P_T \times \{(D_I+D_C)/B_M\} \tag{12}$$

If we want to see whether offloading to a local cloudlet is better than offloading to a remote cloud or not, we have to compute $T(RC)$ - $T(LC)$ and $E(RC)$ – $E(LC)$. By replacing S_C and P_T with S_{CR} and P_{TR} for a remote cloud and S_{CL} and P_{TL} for a local cloudlet in Equations (9) and (10), we can easily see that these two equations become the same as Equations (7) and (8), respectively. This is quite an obvious result because irrespective of whether the mobile node transfers initial data to a local cloudlet or a remote cloud, it pays almost the same cost in terms of elapsed time and consumed energy.

Analysis with the Proposed Model

In this section we use the model derived in the previous section to analyze which location, among a mobile, a local

ISSN:2319-7900

cloudlet, and a remote cloud, is the best place for executing offloadable C_C instructions for two cases as identified as in the two subsections in the previous section. To make the analysis more amenable, we survey data collected from some real mobile nodes, networks, and clouds and choose realistic numbers for some of the symbols used in the equations

- S_M: ARM Cortex A7 processor which is used in many mobile nodes including smartphones and tablet PCs has the instruction execution speed of 2.85GIPS at 1.5GHz and we use this number.
- S_{CR}: Intel Xeon processors are popularly used in server machines and the Xeon 5690 processor has the speed of 84GIPS at 3.46GHz[11]. This number is almost 30 times of the speed of ARM Cortex A7. But a cloud server has much faster memory hierarchy and higher performance for floating point calculation and can provide more numbers of cores to an application. Therefore, in a real situation the speedup of a cloud node over a mobile can easily surpass 100~150.
- S_{CL}: Because a cloudlet can be assembled from the same kind of off-the-shelf server machines as in a remote cloud, the speed of each of component server machine can be almost the same. But a cloudlet will have less number of server machines and the internal physical network connecting them and supporting operating environment software can be slower. Using admission control mechanisms in [13], we can assign almost the same number of cores to an application as in a remote cloud, although the number of simultaneously executable applications will be much lower in a cloudlet. With these observations, we guess that the speedup of a cloud over a cloudlet will not be very high and we can assume around 2~4 and call this speedup factor F in this paper.
- B_M: The bottleneck in the communication path between a mobile node and an Internet server is the wireless link directly connected to the mobile node. WiFi and 3G/4G are popular technologies for the wireless section. But it is well known that 3G/4G has longer delay and consumes more energy than WiFi [14] and, therefore, we assume the use of WiFi. IEEE802.11n has a data rate of 72.2 Mbps with 20 MHz bandwidth. Assuming around 40% throughput, we choose 30Mbps for BM.
- T_{PR}: In [15], over 90% of users experience not greater than 25msec latency to access the closest Amazon cloud center. We choose T_{PR} to be 25msec.
- T_{PL}: Akamai is a content distribution network consisting of over 20,000 hosts and can be much closer to a user. [15] shows that over 90% of users experience not greater than 5msec latency to access the closest Akamai host. But in this paper, we want to be more aggressive assuming that much more cloudlet sites are deployed than Akamai and choose T_{PL} to be around 1msec.

Table 1 shows energy consumption data in watts for some mobile devices. Because faster processors are adopted in more recent mobile nodes and consume more energy during computation and transmission mode we choose energy consumption values as follows

- P_C: 1.0 watt
- P_I: 0.3 watt
- P_T: 2.0 watts

Table 1. Energy Consumption in Mobile Devices

Mobile Devices	P_C	P_I	P_T
HP iPAQ PDA 400MHz [8]	0.9	0.3	1.3
Nokia N810 400MHz [9]	0.8		1.5
Openmoko Neo Freerunner [16]		0.27	
Galaxy S2 1.5GHz [17]		0.36	1.7

A. No Initial Data

For offloading to a local cloudlet to be beneficial, four inequalities $T(MN)$-$T(LC)$ = Equation $(5) > 0$, $E(MN)$-$E(LC)$ = Equations $(6) > 0$, $T(RC)$-$T(LC)$= Equation $(7) > 0$, and $E(RC)$-$E(LC)$ = Equation $(8) > 0$ should be satisfied. But we know that last two inequalities are the same in the previous section. We start with the first inequality.

$$CC(1/S_M\text{-}1/S_{CL})\text{-}2N\times T_{PL}\text{-}D_C/B_M > 0$$

In the above inequality, because $S_{CL} >> S_M$, $1/S_M$-$1/S_{CL}$ can be approximated to $1/S_M$. Then if we divide the inequality by N, it becomes

$$CC'/S_M > 2\times T_{PL}+D_C'/B_M \tag{13}$$

The second inequality becomes

$$C_C\times(P_C/S_M\text{-}P_I/S_C)\text{-}\ P_I\times(2N\times T_P)\text{-}P_T\times(D_C/B_M) > 0$$

Using the fact $P_C/S_M >> P_I/S_C$ and dividing both sides by N, we get

$$P_C\times C_C'/S_M > P_I\times 2T_P+P_T\times D_C'/B_M > 0 \tag{14}$$

From the third inequality, we get

$$C_C\times(1/S_{CR}\text{-}1/S_{CL})+2N\times(T_{PR}\text{-}T_{PL}) > 0$$

By dividing both sides by N we obtain

ISSN:2319-7900

$$2(T_{PR}\text{-}T_{PL}) > C_C'\times(1/S_{CL}\text{-}1/S_{CR}) = C_C'\times(F\text{-}1)/S_{CR} \quad (15)$$

After inserting parameter values assumed in the beginning of this section into Equations (13), (14), and (15), we obtain the following three inequalities

$$C_C' > 5.7\times10^6+95\times D_C' \quad (16)$$
$$C_C' > 1.7\times10^6+190\times D_C' \quad (17)$$
$$C_C' < 4\times10^9/(F\text{-}1) \quad (18)$$

We see that C_C' is lower-bounded by Equations (16) and (17) and upper-bounded by Equation (18).

For offloading to a remote cloud to be beneficial, four inequalities $T(MN)$-$T(RC)$ = (5) > 0, $E(MN)$ –$E(RC)$ = Equation (6) > 0, $T(RC)$-$T(LC)$ = Equation (7) < 0, and $E(RC)$-E(LC) = Equation (8) < 0 should be satisfied. Following the similar calculations we obtain following three inequalities.

$$C_C' > 1.43\times10^8+95\times D_C' \quad (19)$$
$$C_C' > 4.37\times10^7+190\times D_C' \quad (20)$$
$$C_C' > 4\times10^9/(F\text{-}1) \quad (21)$$

We see that for offloading to a remote cloud to be beneficial, more instructions should be executed during the process step.

From Equations (16) to (18), we see that if the process step has a computation requirement large enough to compensate for the cost needed to exchange data between a mobile node and a local cloudlet, it is better to execute the process step on the local cloudlet. But From Equations (19) to (21), if the computation requirement of a process step gets even larger, then it is better to offload the process step to a remote cloud.

B. Initial Data at a Mobile Node

If the initial data has a modest size, it can be stored at a mobile node. The conditions that must be satisfied to make offloading to a local cloud the best decision are T(MN)-$T(LC)$ = Equation (11) > 0, $E(MN)$-$E(LC)$ = Equation (12) > 0, $T(RC)$-$T(LC)$ = Equation (7) > 0, and $E(RC)$-$E(LC)$ = Equation (8) > 0. Following the same approximations and calculations, we obtain the following inequalities.

$$C_C' > 5.7\times10^6+95\times D_C'+95\times(D_I/N) \quad (22)$$
$$C_C' > 1.7\times10^6+190\times D_C'+190\times(D_I/N) \quad (23)$$
$$C_C' < 4\times10^9/(F\text{-}1) \quad (24)$$

If the conditions $T(MN)$-$T(RC)$ = Equation (11) > 0, $E(MN)$-$E(RC)$ = Equation (12) > 0, $T(RC)$-$T(LC)$ = Equation (7) < 0, and $E(RC)$-$E(LC)$ = Equation (8) < 0, the remote cloud becomes the best place which offload the process step computation to. With the same methods, we get the following inequalities.

$$C_C' > 1.43\times10^8+95\times D_C' +95\times(D_I/N) \quad (25)$$
$$C_C' > 4.37\times10^7+190\times D_C' +190\times(D_I/N) \quad (26)$$
$$C_C' > 4\times10^9/(F\text{-}1) \quad (27)$$

We see that to make offloading useful, the process step should have more instructions to offset the extra cost to transfer the initial data to a local cloudlet or a remote cloud. But this extra cost becomes smaller as more interactions are made with a user.

Conclusion

We consider the execution of an interactive application in a mobile cloud computing environment including a local cloudlet and remote cloud. We consider two cases for an interactive application: without initial data and with initial data stored at a mobile device. We also assume that the program consists of two parts: one that should be executed at a mobile device and the other that can be offloaded to an outside node, either a local cloudlet or a remote cloud. We introduce models for computing completion time and energy consumption of an interactive application when its offloadable portion is executed on a mobile device, a local cloudlet, or a remote cloud. Applying practical numbers to some of the model parameters such as processing speed, bandwidth, propagation delay, and energy usage of a mobile device, we derive conditions under which computation offloading to either a local cloudlet or a remote cloud becomes beneficial in terms of completion time and energy consumption

Acknowledgments

This work was supported by 2013 Hongik University Research Fund.

References

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," NIST Special Publication 800-15.

[2] A. Li, X. Yang, S. Kandula, and M. Zhang, "CloudCmp: Comparing Public cloud Providers," ACM Internet Measurement Conference, 2010.

[3] G.Lu and W.Zeng, "Cloud Computing Survey," Applied Mechanics and Materials, Vols 530-531, pp.650-661, 2014.

ISSN:2319-7900

[4] N. Fernando, S.W. Loke, and W. Rahayu, “Mobile Cloud Computing: A Survey,” Future Generation Computer Systems, Vol. 29, pp. 84-106, 2013.

[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, 2011.

[6] L. Guan, X. Ke, M. Song, and J. Song, “A Survey of Research on Mobile Cloud Computing,” 10th IEEE/ACIS international Conference on Computer and Information Science, 2011.

[7] M. Satyanarayanan, P.Bahl, R. Caceres, and N. Davies, “The Case for VM-based Cloudlets in Mobile Computing,” IEEE Pervasive Computing, Vol. 8, No. 4, pp. 1423, 2009.

[8] K. Kumar and Y.-H. Lu, “Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?” IEEE Computer, pp. 51-56, April 2010.

[9] A.P. Miettinen and J.K. Nurminen, “Energy Efficiency of Mobile Clients in Cloud Computing,:” The 2nd USENIX Conference, 2010.

[10] B.-G. Chun, S. Khm, P. Maniatis, M. Naik, and A. Patti, “CloneCloud: Elastic Execution between Mobile Device and Cloud,” ACM Symposium on Computer Systems (EuroSys), pp. 301-314, 2011.

[11] E. Cuervo, A. Balasubramanian, D.-K. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, “Maui: Making smartphone Last Longer with Code Offload,” The 8th International Conference on Mobile Systems, Applications, and Services (MobisSys’10), 2010.

[12] N. Sorensen, Industry Benchmarks Performance, http://www.cisco.com.

[13] D.T. Hoang, D. Niyato, and P. Wang, “Optimal Admission Control Policy for Mobile Cloud Computing Hotspot with Cloudlet,” IEEE Wireless Communications and Networking Conference: Services, Applications, and Business, 2012.

[14] N. Balasubramanian, A. Balasubramanian, and A. Vekataramani, “Energy Consumption in Mobile Phones: a Measurement Study and Implications for Network Applications,” The 9th ACM SIGCOMM Conference on Internet Measurement (IMC’09), 2009.

[15] J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, and V. Sekar, “Making Middleboxes Someone Else’s Problem: Network Processing as a Cloud Service,” ACM SIGCOMM Conference, 2012.

[16] A. Carroll and G. Heiser, “An Analysis of Power Consumption in a Smartphone,” USENIX Annual Technical Conference, 2010.

[17] M.Y. Malik, “Power Consumption Analysis of a Modern Smartphone,” Lecture Notes in Computer Science, 2012.

Biographies

YOUNG-CHUL SHIM received the B.S. degree in Electronic Engineering from the Seoul National University, Korea, the M.S. degree in Electrical Engineering from KAIST, Korea, and the Ph.D. degree in Computer Science from the University of California, Berkeley, U.S.A. Currently he is a Professor of Computer Engineering at Hongik University. His teaching and research areas include network protocols, network security, and cloud computing.