International Journal of Advanced Computer Technology (IJACT)

ISSN:2319-7900

EFFECTS OF CLOUDLETS ON INTERACTIVE 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. A user can execute the application completely on his mobile device or move the offloadable portion of the application either to a local cloudlet or a remote cloud center. We assume that local cloudlet is accessible through a wireless mesh network and the remote cloud center is accessed using LTE. We derive models for calculating the completion time and energy consumption for each case. Applying typical values for parameters in the model, we analyze and derive conditions under which offloading to a local cloudlet becomes the most beneficial in terms of both completion time and energy consumption. During the analysis we use two models for the user mobility: linear mobility and geometric mobility.

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 are 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 collection, 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.

Popular cloud centers such as Amazon AWS, Microsoft Azure, and Google AppEngine are deployed only on a selected small number of locations and they may be located far away from a mobile user and the communication latency to them can become non-negligible. If running an application requires a large number of interactions, this non-negligible latency may produce a negative effect on the response time and battery power saving. Recently the concept of cloudlets was introduced. 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 cloud center and deploy a large number of cloudlets.

ISSN:2319-7900

Mobile devices can have various kinds of interfaces for accessing wireless networks: LTE, WiMAX, WiFi, Bluetooth, etc. LTE is a 3.9G wireless access interface for mobile devices. It supports the maximum mobility but has high latency to access Internet. WiMAX can provide maximum mobility and high performance access to Internet for mobile devices but its latency is shown to be higher than LTE and, therefore, is not considered in this paper [8]. WiFi provides wide bandwidth and has smaller latency but most of the currently deployed WiFi networks do not support user mobility. If a user leaves the communication range of an AP(Access Point), it loses the connection through this AP and should make a new connection through a new AP. Bluetooth's bandwidth is too narrow and, therefore, is not considered in this paper.

In this paper we consider a user who moves with a mobile device and executes an interactive application. He can execute all or part of the application on the mobile device, a local cloudlet, or a remote cloud. When he uses the remote cloud, we assume that he accesses the Internet through LTE which provides the maximum mobility. When he uses the local cloudlet, we assume that he accesses the local cloudlet through a wireless mesh network as in Figure 1. In the figure an access router (AR) provides WiFi access to a mobile device (MD) and is co-located with a local cloudlet server. An AR has a WiFi access to neighboring ARs and together they make a wireless mesh network [9]. A mobile device within the communication range of AR1 can use the cloudlet server co-located with AR1. When the mobile device moves to the range of AR2, it can still use the same cloudlet server because AR2 can relay the messages to and from the cloudlet server at AR1 using the wireless interface between AR1 and AR2.

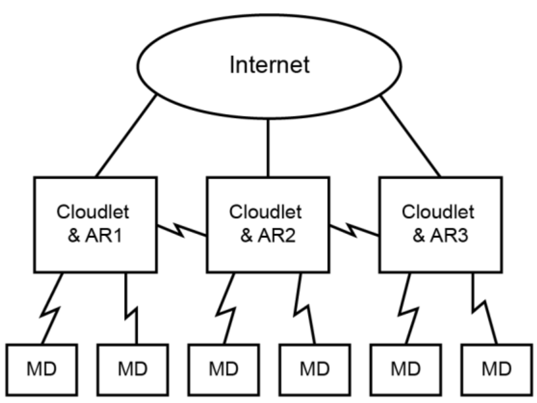

Figure 1. Cloudlets within a Wireless Mesh Network

In this paper we consider interactive applications running in a mobile cloud computing environment including both local cloudlets and remote clouds. A user can run the interactive application on his mobile device, a local cloudlet connected through a wireless mesh network, or a remote cloud connected through LTE. 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 the typical values to the parameters in the model, we derive conditions under which offloading to local cloudlets becomes the most beneficial. We assume that mobile devices move in this paper.

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 describe the conditions under which offloading to local cloudlets become the most 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 to 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.

After the initial data loading phase, which is performed only when required, comes the interactive computation phase. In the interactive computation phase, an activity con-

ISSN:2319-7900

sisting of three steps is 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 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 [10].

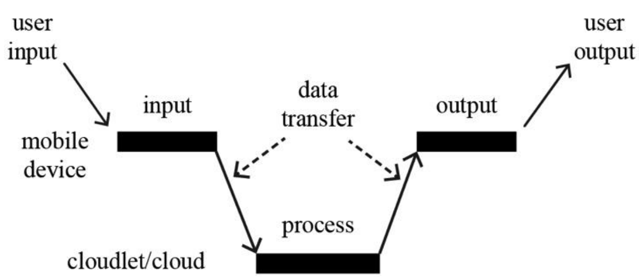

Figure 2. Remote Execution of a Process Step

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 previous 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 computation 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 exchanging data with a mobile node.

There can be two kinds of interactive applications: one which does not require initial data and the other which requires initial data. In this paper we consider only the applications which do not require initial data for simplicity.

Before we present the models for completion time and energy consumption, we 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 $C_C/N = C_C'$ 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 S_{CL} for the speed of a local cloudlet and S_{CR} for a remote cloud.
- F: Speedup of a remote cloud over a local cloudlet defined to be S_{CR}/S_{CL}.
- D: 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 $D/N = D'$.
- B: The bandwidth between a mobile and an outside node. If we have to distinguish the bandwidth with a local cloudlet from the bandwidth with a remote cloud, we use B_L for a local cloudlet and B_R for a remote cloud.
- 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 T_{PL} for a local cloudlet and T_{PR} 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.

ISSN:2319-7900

- 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.
- *MN*: Mobile node. In this we use two terms a mobile node and a mobile device interchangeably.
- *LC*: Local cloudlet
- *RC*: Remote cloud

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

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

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. 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

$$T(LC \text{ or } RC) = C_M/S_M+C_C/S_C+2N\times T_P+D/B \quad (3)$$

$$E(LC \text{ or } RC) = P_C\times(C_M/S_M)+P_I\times(C_C/S_C+2N\times T_P) +P_T\times(D/B) \quad (4)$$

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, T_P and B become S_{CL}, T_{PL}, and B_L if a local cloudlet is used and become S_{CR}, T_{PR}, and B_R if a remote cloud is used.

First we consider the case when all the computations are made on a mobile device and the case when C_C instructions are offloaded to a local cloudlet. To compare the completion time and consumed energy between these two cases, we compute $T(MN) - T(LC)$ and $E(MN) - E(LC)$ as follows. In the computation we assume that $S_{CL} >> S_M$.

$$T(MN)-T(LC) = C_C(1/S_M-1/S_{CL})-2N\times T_{PL}-D/B_L \approx C_C/S_M-2N\times T_{PL}-D/B_L = N\times\{C_C'/S_M-2T_{PL}-D'/B_L\} \quad (5)$$

$$E(MN)-E(LC) = C_C\times(P_C/S_M-P_I/S_{CL})- P_I\times(2N\times T_{PL})-P_T\times(D/B_L) \approx C_C\times P_C/S_M-2N\times P_I\times T_{PL}-P_T\times D/B_L = N\times\{P_C\times C_C'/S_M-2T_I\times T_{PL}-P_T\times D'/B_L\} \quad (6)$$

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

$$T(RC)-T(LC) = C_C\times(1/S_{CR}-1/S_{CL})+2N\times(T_{PR}-T_{PL})+D\times(1/B_R-1/B_L) = N\times\{C_C'\times(1-F)/S_{CR}+2(T_{PR}-T_{PL})+D'\times(1/B_R-1/B_L)\} \quad (7)$$

$$E(RC)-E(LC) = P_I\times\{C_C(1/S_{CR}-1/S_{CL})+2N\times(T_{PR}-T_{PL})\} +P_T\times D\times(1/B_R-1/B_L) = N\times[P_I\times\{C_C'\times(1\text{-F})/S_{CR}+2(T_{PR}-T_{PL})\} +P_T\times D'\times(1/B_R-1/B_L)] \quad (8)$$

Executing the offloadable instructions on a local cloudlet becomes the most beneficial, when all the equations from (5) to (8) become greater than 0. From this we obtain the following 4 inequalities.

$$C_C' > S_M\times(2T_{PL}+D'/B_L) \quad (9)$$
$$C_C' > S_M\times(2P_I\times T_{PL}+P_T\times D'/B_L) \quad (10)$$
$$C_C' < S_{CR}\times\{2(T_{PR}-T_{PL})+D'\times(1/B_R-1/B_L)\}/(F-1) \quad (11)$$
$$C_C' < S_{CR}\times\{2(T_{PR}-T_{PL})+(P_T/P_I)\times D'\times(1/B_R-1/B_L)\}/(F-1) \quad (12)$$

Analysis Results

In this section we derive conditions under which offloading to a local cloudlet through a wireless mesh network is more advantageous in terms of completion time and energy consumption than both offloading to a remote cloud through LTE and executing the application only on a mobile device.

We first explain the case of offloading to a local cloudlet through a wireless mesh network with Figure 1 in more detail. Let's assume that a mobile user initiates an interactive application with his mobile device within the range of the access router *AR1*. Then the cloudlet co-located with *AR1* is the closest cloudlet from the mobile device and, therefore, the offloadable part of the application is offloaded to that cloudlet. If the user moves to the range of *AR2*, the user has two choices for the cloudlet. He can use the previous cloudlet or he can migrate the virtual machine on the previous cloudlet to the new cloudlet co-located with *AR2*. In this paper we do not consider the migration of a virtual machine

ISSN:2319-7900

because the live migration of a virtual machine from one physical machine to another physical machine, which does not belong to the same cloud center comprising an extremely fast local area network and a special purpose middleware, takes several hundred seconds [11]-[12]. So even though the user moves from the range of *AR1* to *AR2* and then to *AR3*, the mobile device of the user communicates with the virtual machine originally deployed on the cloudlet co-located with *AR1*. If a user is located in the range of *AR3*, the messages from the mobile device is delivered to *AR3*, then to *AR2* through the wireless interface and then finally to *AR1* through the wireless interface as the access routers make a wireless mesh network. As the user moves away from the initial access router, the bandwidth available to the user between the mobile device and the physical machine running the virtual machine decreases and the latency that the user experiences increases. It is known that bandwidth is reduced to half as one hop of wireless interface is crossed [13] and the latency increases linearly as a wireless interface is crossed.

For the analysis we consider two mobility models for mobile users. In the first model, called a linear mobility model, if a user visits the ranges of N access routers, then he stays in each range during the same amount of time. For example, if in Figure 1 a mobile user moves among ranges of *AR1*, *AR2*, and *AR3* for total 30 seconds, then he stays for 10 seconds at the range of each access routers. In the second model, called a geometric mobility model, if a user stays in a range which is N hops away from the initial range for T seconds, then he stays for $T/2$ seconds in a range which is $N+1$ hops away. For example, let's assume that in Figure 1 a mobile user starts from the range of *AR1* and visits the ranges of *AR2* and *AR3*. If he stays at the range of *AR1* for 8 seconds, he stays at the ranges of *AR2* and *AR3* for 4 seconds and 2 seconds, respectively.

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[14]. 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 [15], 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 assume around 2~16 and call this speedup factor F in this paper.
- B_R: A remote cloud is accessed through LTE. From the many experiments conducted in the U.S.A. the largest bandwidth for LTE was 20Mbps [16].
- B_L: A local cloudlet is accessed through a wireless mesh network. We assume that an access router uses WiFi to access mobile devices and neighboring access routers. IEEE802.11n has a bandwidth of 72.2 Mbps using 20 MHz. Assuming around 40% throughput, we choose 30Mbps. If a mobile user accesses to a cloudlet through one access router (one hop away), he can use 30Mbps. But if he is N hops away from a cloudlet, he can use $30*(1/2)^{N-1}$ Mbps.
- T_{PR}: In [17], over 90% of users experience not greater than 25msec latency to access the closest Amazon cloud center through Internet. And it is shown the Internet access latency for LTE is almost 100msec [18]. So we choose T_{PR} to be 120msec.
- T_{PL}: Although the latency of WiFi has high variability we assume that one hop latency for WiFi is 20msec in this paper [19, 20]. So if a mobile user is N hops away from a cloudlet in the wireless mesh network, he experiences $20*N$ msec latency.

Table 1. Energy Consumption in Mobile Devices

Mobile Devices	P_C	P_I	P_T
HP iPAQ PDA 400MHz [21]	0.9	0.3	1.3
Nokia N810 400MHz [22]	0.8		1.5
Openmoko Neo Freerunner [23]		0.27	
Galaxy S2 1.5GHz [24]		0.36	1.7

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

ISSN:2319-7900

Table 2. Lower Bound and Upper Bound for C_C' (Offloadable Computation in One Input-Process-Output Step) Assuming a Linear Mobility Model

Speedup (F)	# of Hops	Data Size				
		1k	10k	100k	1M	10M
2	1	0.11, 16.80	0.11, 16.81	0.12, 16.94	0.22, 18.20	1.93, 30.80
	2	0.17, 15.12	0.17, 15.13	0.18, 15.26	0.30, 16.52	2.58, 29.10
	3	0.23, 13.44	0.23, 13.40	0.24, 13.05	0.39, 9.44	-
	4	0.29, 11.75	0.29, 11.65	0.31,10.58	-	-
	5	0.34, 10.06	0.34, 9.88	0.37, 8.06	-	-
	6	0.40, 8.37	0.40, 8.11	0.43, 5.51	-	-
	7	0.46, 6.68	0.46, 6.34	0.49, 2.92	-	-
	8	0.51, 4.99	0.52, 4.57	-	-	-
	9	0.57, 3.30	0.57, 2.80	-	-	-
	10	0.63, 1.61	0.63, 1.03	-	-	-
4	1	0.11, 5.60	0.11, 5.60	0.12, 5.65	0.22, 6.07	1.93, 10.27
	2	0.17, 5.04	0.17, 5.04	0.18, 5.09	0.30, 5.51	2.58, 9.71
	3	0.23, 4.48	0.23, 4.47	0.24, 4.35	0.39, 3.15	-
	4	0.29, 3.92	0.29, 3.88	0.31, 3.53	-	-
	5	0.34, 3.36	0.34, 3.30	0.37, 2.67	-	-
	6	0.40, 2.79	0.40, 2.73	0.43, 1.84	-	-
	7	0.46, 2.23	0.46, 2.11	0.49, 0.97	-	-
	8	0.51, 1.66	0.52, 1.52	-	-	-
	9	0.57, 1.100	0.57, 0.93	-	-	-
	10	-	-	-	-	-
8	1	0.11, 2.40	0.11, 2.40	0.12, 2.42	0.22, 2.60	1.93, 4.40
	2	0.17, 2.16	0.17, 2.16	0.18, 2.18	0.30, 2.36	2.58, 4.16
	3	0.23, 1.92	0.23, 1.91	0.24, 1.86	0.39, 1.35	-
	4	0.29, 1.68	0.29, 1.66	0.31, 1.51	-	-
	5	0.34, 1.44	0.34, 1.41	0.37, 1.15	-	-
	6	0.40, 1.12	0.40, 1.16	0.43, 0.79	-	-
	7	0.46, 0.95	0.46, 0.91	-	-	-
	8	0.51, 0.71	0.52, 0.65	-	-	-
	9	-	-	-	-	-
	10	-	-	-	-	-
16	1	0.11, 1.12	0.11, 1.12	0.12, 1.13	0.22, 1.21	1.93, 2.05
	2	0.23, 0.90	0.23, 0.89	0.18, 1.02	0.30, 1.10	-
	3	0.23, 0.90	0.23, 0.89	0.24, 0.87	0.39, 0.63	-
	4	0.29, 0.78	0.29, 0.78	0.31, 0.71	-	-
	5	0.34, 0.67	0.34, 0.66	0.37, 0.54	-	-
	6	0.40, 0.56	0.50, 0.54	-	-	-
	7	-	-	-	-	-
	8	-	-	-	-	-
	9	-	-	-	-	-

	10	-	-	-	-	-

Table 3. Range Within Which a Mobile User Can Move in Number of Hops (Assuming a Linear Mobility Model)

Speedup	Data Size (Bits)				
	1k	10k	100k	1M	10M
2	> 10	> 10	7	3	2
4	9	9	7	3	2
8	8	8	6	3	2
16	6	6	5	3	1

Table 4. Lower Bound and Upper Bound for C_C' (Offloadable Computation in One Input-Process-Output Step) Assuming a Geometric Mobility Model

Speedup(F)	# of Hops	Data Size				
		1k	10k	100k	1M	10M
2	1	0.11, 16.80	0.11, 16.81	0.12, 16.94	0.22, 18.20	1.93, 30.80
	2	0.15, 15.68	0.15, 15.69	0.16, 15.82	0.27, 17.08	2.33, 29.68
	3	0.18, 14.88	0.18, 14.89	0.19, 15.02	0.31, 16.28	2.59, 28.88
	4	0.20, 14.31	0.20, 14.33	0.21, 14.48	0.33, 15.74	2.74, 28.34
	5	0.21, 13.98	0.21, 13.99	0.22, 14.07	0.35, 14.83	2.83, 22.47
4	1	0.11, 13.98	0.11, 5.60	0.12, 5.65	0.22, 6.07	1.93, 10.27
	2	0.15, 5.23	0.15, 5.23	0.16, 5.27	0.27, 5.69	2.33, 9.89
	3	0.18, 4.96	0.18, 4.96	0.19, 5.01	0.31, 5.43	2.59, 9.63
	4	0.20, 5.78	0.20, 4.78	0.21, 4.83	0.33, 5.25	2.74, 9.45
	5	0.21, 4.66	0.21, 4.66	0.22, 4.69	0.35, 4.94	2.83, 7.49
8	1	0.11, 2.40	0.11, 2.40	0.12, 2.42	0.22, 2.60	1.93, 4.40
	2	0.15, 2.24	0.15, 2.24	0.16, 2.26	0.27, 2.44	2.33, 4.24
	3	0.17, 2.12	0.18, 2.13	0.19, 2.15	0.31, 2.33	2.59, 4.13
	4	0.20, 2.05	0.20, 2.05	0.21, 2.07	0.33, 2.55	2.74, 4.05
	5	0.21, 2.00	0.21, 2.00	0.22, 2.01	0.35, 2.12	2.83, 3.21
16	1	0.11, 1.12	0.11, 1.12	0.12, 1.13	0.22, 1.21	1.93, 2.05
	2	0.15, 1.05	0.15, 1.05	0.16, 1.05	0.27, 1.14	-
	3	0.18, 0.99	0.18, 0.99	0.19, 1.00	0.31, 1.09	-
	4	0.20, 0.96	0.20, 0.97	0.21, 0.97	0.33, 1.05	-
	5	0.21, 0.93	0.21, 0.93	0.22, 0.94	0.35, 0.99	-

ISSN:2319-7900

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

Executing the offloadable instructions on a local cloudlet becomes the most beneficial, when all the 4 inequalities (9) to (12) are satisfied. (9) and (10) define the lower bound for the size of the offloadable portion of the computation in one input-process-output step (C_C'), in the number of instructions. (11) and (12) define the upper bound.

We applied the parameter values explained above to (9)-(12) and obtained the size of the offloadable portion of the computation (C_C') as in Tables 2 and 4. Table 2 assumes the linear mobility model and Table 4 assumes the geometric mobility model. In the tables we consider speedup factors 2, 4, 8, and 16. For data exchanged during one input-process-output phase (D'), we consider the sizes 1k, 10k, 100k, 1M, and 10M bits. The number of hops represents the range of the user's movement. If it is 1, the user does not leave the range of the initial access router. If it is 5, the user moves as far as 4 hops away from the range of the initial access router. The values represented as a pair of two numbers are the lower and the upper bounds for the size of C_C' in giga instructions. If the lower and upper bound entry is filled with a hyphen, it means that the lower bound is bigger than the upper bound and that particular entry is not possible.

For the linear mobility model, as the data size increases, the lower bound increases and the speedup has no effect. The upper bound constantly increases with a small number of hops but increases and then decreases with a large number of hops. Increasing speedup decreases the upper bound. As the number of hops increases, the lower bound increases with no effect from the speedup and decreases the upper bound with further decreasing effect from the increasing speedup. Table 3 summarizes Table 2 and shows how far a user can move away from the initial access router in the number of hops. Table 3 shows as either the speedup or the data size increases, the range in which a user can move around decreases. Particularly if the speedup is 16 and the data size is 10 Mbits, then the user cannot leave the range of the initial access router.

From Table 4, we can see that most of the analysis results for the linear mobility model apply to the geometric mobility model except one fact that as the data size increases, the upper bound for the C_C' size constantly increases. Although it is not shown in Table 4, in our analysis we increased the number of hops to 10, and we still could find the valid values for the lower and upper bounds except when the data size is 10Mbits. This means that if the exchanged data size is not excessively large, the user can move very far away from the initial access router.

Conclusion

We considered the execution of an interactive application in a mobile cloud computing environment. A user can execute the application completely on his mobile device or move the offloadable portion of the application either to a local cloudlet or a remote cloud center. We assume that local cloudlet is accessible through a wireless mesh network using WiFi and the remote cloud center is accessed using LTE. We derived models for calculating the completion time and energy consumption for each case. Applying typical values for the model parameters including the computation speed of a computing node (mobile device, local cloudlet, and a remote cloud), bandwidth of a wireless interface, and the latency to an outside computing node (a local cloudlet and a remote cloud), we analyzed and derived conditions under which offloading to local cloudlet is the most beneficial in terms of both completion time and energy consumption. During the analysis we used two models for the user mobility: linear mobility and geometric mobility. We varied three parameters values (the speedup factor of a remote cloud over a local cloudlet, the size of data exchanged between a mobile device and either a local cloudlet or a remote server, and the range within which a user can move) and analyzed how these three parameters affect the conditions under which offloading to be most beneficial.

Acknowledgments

This work was supported by 2014 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.
[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, Ap-

ISSN:2319-7900

plications, 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] Yi-Bing Lin et al "Performance Measurement of TD-LTE, WiMAX and 3G Systems," IEEE Wireless Communications, vol. 20, no. 3 pp. 153-160, June 2013.
[9] I.F. Akyildiz and S. Wang, "A Survey on Wireless Mesh Network," IEEE Radio Communications, pp. S23-S30, September 2005.
[10] Y.-C. Shim, "Modeling and Analysis of Completion Time and Energy consumption of Applications in Mobile Cloud Computing Environments," Int. Journal of Advanced Computer Technology, vol. 3, no. 6, pp. 60-66, 2014.
[11] R. Bradford, E. Kotsovinos, A. Feldmann, and H. Schioberg, "Live Wide-Area Migration of Virtual Machines including Local Persistent State," VEE'07, June 2007.
[12] E. Harney, S. Goasguen, J. Martin, M. Murphy, and M. Westall, "The Efficacy of Live Virtual Machine Migrations over the Internet," VTDC'07, November 2007.
[13] CloudTrax, CloudTrax (A Cloud-based Network Controller) Network Planning Guide, 2012.
[14] N. Sorensen, Industry Benchmarks Performance, http://www.cisco.com.
[15] 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.
[16] AT&T LTE Network Tops Speed Test, but Verizon Still Has the Best LTE Coverage, http://bgr.com/2013/03/11/4g-network-speeds-368339.
[17] 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.
[18] 3G/4G wireless Network Latency: How do Verizon, AT&T, Sprint and T-Mobile Compare, http://www.fiercewireless.com/special-reports/3g-4g-wireless-network-latency-how-do-verizon-att-sprint-and-t-mobile-compar.
[19] WiFi Direct Data Performance Tests, http://msdn.microsoft.com/en-us/library/windows/hardware/dn247504.aspx.
[20] P. Payaswini and D.H. Manjaiah, "Simulation and Performance Analysis of Vertical Handoff between WiFi and WiMAX using Media Independent Handover Services," International Journal of Computer Applications, vol. 87, no. 4, Feb. 2014.
[21] K. Kumar and Y.-H. Lu, "Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?" IEEE Computer, pp. 51-56, April 2010.
[22] 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.
[23] A. Carroll and G. Heiser, "An Analysis of Power Consumption in a Smartphone," USENIX Annual Technical Conference, 2010.
[24] 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. Professor Shim may be reached at shim@cs.hongik.ac.kr.