CellCloud: A Novel Cost Effective Formation of Mobile Cloud Based on Bidding Incentives

Shahid Al Noor, Ragib Hasan, and Md Munirul Haque
{shaahid, ragib, mhaque}@cis.uab.edu
Department of Computer and Information Sciences
University of Alabama at Birmingham
Birmingham, AL 35294-1170

***Abstract*—Cloud computing has become the dominant computing paradigm in recent years. As clouds evolved, researchers have explored the possibility of building clouds out of loosely associated mobile computing devices. However, most such efforts failed due to the lack of a proper incentive model for the mobile device owners. In this paper, we propose CellCloud – a practical mobile cloud architecture which can be easily deployed on existing cellular phone network infrastructure. It is based on a novel reputation-based economic incentive model in order to compensate the phone owners for the use of their phones as cloud computing nodes. CellCloud offers a practical model for performing cloud operations, with lower costs compared to a traditional cloud. We provide an elaborate analysis of the model with security and economic incentives as major focus. Along with a cost equation model, we discuss detailed results to prove the feasibility of our proposed model. Our simulation results show that CellCloud creates a win-win scenario for all three stakeholders (client, cloud provider, and mobile device owners) to ensure the formation of a successful mobile cloud architecture.**

***Keywords*-mobile cloud; bidding; challenges; trustworthiness; cost model;**

I. Introduction

Cloud computing is a popular computing model due to its low-cost, scalability, and high-performance. However, in some cases, there are hidden costs of cloud computing which make it infeasible compared to private hosting [1], [2]. From the operational and structural point of view, the fixed structure of cloud data centers can cause underutilization of resources if there is a rapid decrease in clients' demands for cloud services. Recently, major cloud service providers including Amazon and Microsoft failed to earn expected revenues due to the unexpected shutdown of government budget [3]. The primary reason of this loss is due to the lack of flexibility to contract and expand the resources based on the client requirements. However, if the servers themselves could be outsourced from the cloud service providers to individuals with excess resources, we could design a cloud service that is not subject to underutilization of resources. Using mobile devices, it is possible to form a highly scalable ad hoc mobile cloud with low infrastructure set up cost and time. Hence, mobile cloud computing is introduced where alike a traditional cloud, a virtualized interface is formed using mobile devices.

Researchers have defined mobile cloud from two aspects. According to the first aspect, mobile cloud computing is an infrastructure where mobile users use backend cloud system for storing and processing data required to run an application [4]. The second aspect contends that mobile cloud computing enhances the storage and computational power of the cloud system by using the unused resources of mobile devices [5]. In this paper, we would like to focus on the second aspect of mobile cloud. Some applications use mobile sensed data, which is both time consuming and expensive to send to the traditional cloud for processing. A more effective approach in such scenarios will be processing data locally using the mobile cloud as addressed in the second aspect. Another big motivation of preferring the second aspect of mobile cloud is the availability of millions of unused mobile devices. According to the survey of Lockout Inc, around 20, 16, and 19 percent of the people have one, two, and more than two unused mobile devices respectively [6].

There are several reasons to prefer a mobile cloud over traditional cloud. Firstly, a mobile cloud requires low set up and maintenance costs as compared to traditional cloud. Secondly, a mobile cloud can be expanded easily keeping pace with the growing demand of the clients. Thirdly, tasks can be easily distributed and transferred among mobile devices as necessary since the infrastructure of mobile network is already available. Finally, from the work by Alzain et al. [7], Bendahmane et al. [8], and Lagar et al. [9], we can conclude that keeping higher redundancy in task computation ensures more authentic results. Therefore, the probability of getting legitimate result is higher in a mobile cloud than a traditional cloud as we have more unused mobile devices which can be used for the redundant computation of a single task.

Most of the research on mobile cloud fall under the first domain focusing on using the cloud in the backend to enhance the storage and computational power, battery longevity, safety, and security of mobile device [10], [11], [12], [13]. On the contrary, very few researchers considered the use of mobile devices as an integral part of a cloud [14], [15]. Low storage and computing power of the unused mobile devices were the biggest stumbling blocks for researchers to exploit the opportunity of forming mobile clouds with these devices. However, with the advance of technology, researchers have just started considering the second approach where mobile devices are used as integral part of a cloud [14], [15]. Though these models [14], [15] include an architecture for forming a mobile cloud, the absence of appropriate cost/incentive model fails to motivate the owners to participate. These models also do not address the feasibility of utilizing these unused mobile devices from the provider's perspective. To solve this, we introduce *CellCloud* – a practical mobile cloud architecture which can be easily deployed on existing cellular phone network infrastructure. In CellCloud, we address these issues by exploring a bidding strategy for providing incentives to mobile device owners, and also quantify the benefits achieved by cloud providers by using mobile devices as cloud nodes.

In CellCloud, mobile devices known as bidders are hired following a bidding process. During the bidding process, each bidder is offered monetary incentive based on their available resource and rating point. The rating point determines the trustworthiness of the bidder for a particular task. Based on client requirement, CellCloud provider hires the required number of bidders for a task, divides the task into smaller subtasks, and distributes them among bidders for computation. The CellCloud provider uses Map Reduce based scheme for its computations. The overall model proves to be cost effective for both cloud providers and mobile owners creating a win-win situation for all stake holders.

Contributions:

1) To the best of our knowledge, CellCloud is the very first attempt to form a mobile cloud using the network of the mobile operators and unused mobile phones. We introduce a bidding process for forming the mobile cloud where mobile device owners can submit their free resources during bidding and get incentives for their resources.
2) We introduce the novel concept of rating point associated with the bidders (mobile devices) to ensure trustworthiness and reliability of the service in hand. Moreover our model enables users to choose different service cost based on the rating level of the computing nodes providing a unique opportunity to the clients.

The rest of the paper is organized as follows. Section II describes the motivation of our research work. In section III, we discuss some of the related works on mobile cloud. Section IV introduces our mobile cloud architecture. We describe the strategies for assigning rating point, measuring cost of the operation, selecting base stations and bidders in section V, VI, VII and VIII respectively. Section IX elaborates possible challenges in mobile cloud followed by experimental results in section X. Finally we conclude with discussion and future directions in section XI and XII respectively.

II. Motivation

Mobile cloud computing requires significantly smaller amount of initial set up cost as compared to a traditional cloud system. The primary reason of this low set up cost is that, it includes the unused mobile devices in cloud platform which is very easy to find. Lockout finds that, approximately 52 percent mobile users intend to donate their old sets for charitable use [6]. Moreover, it is also observed that, mobile devices are in idle state for 89% of the time in a day and during that time the devices use less than 11% of total CPU power [16]. Therefore, with no obvious harm and associated economic benefit, we believe that it would be very easy to motivate the mobile device owners to share their unused mobile devices on cloud platform.

Scalability of client is another big reason to choose mobile cloud than traditional cloud system. Sometimes there can be more demand for resources than expected. In general, a good amount of resources remain unused in public cloud as they want to be on the safe side. Therefore, if at any time there is a certain decrease in client demand, some resources will be unused. Recently amazon, microsoft and some other cloud services failed to earn expected revenues due to the unexpected shutdown of government budget for the conflict arise on US legislation [3]. On the other hand, in our CellCloud architecture, we always hire bidders on demand basis. Since the infrastructure of mobile network is already there, the cost for keeping continuous connection between the bidders and the base stations is almost negligible. We have always more number of reserve bidders. Therefore, our CellCloud architecture will be able to handle both the sudden increase and decrease of clients without any financial downfall.

Mobile cloud can reduce the computational cost significantly as compared to the traditional backend cloud system. Mahesri and Vardhan showed that the average power consumption of a personal computer with 1.3 GHz processor in idle state is 13.13W [17]. Whereas, in idle state, the average power consumption of a mobile phone with 400 MHz processor is only 268.8mW [18]. Hence, total power consumption in mobile cloud is much smaller than traditional cloud. Therefore, we can include more number of bidders in our CellCloud to achieve the same computing capabilities as provided by traditional cloud. This comparatively higher power consumption issue also plays a big role in higher operational cost for traditional cloud services.

Ensuring trustworthiness during computation is another major reason to choose mobile cloud over traditional cloud system. Bendahmane et al. discussed two popular methods for ensuring the authenticity in cloud computation; majority based voting and m-first voting system [8]. However, both of these methods use multiple virtual devices for computing a single task. We use a similar approach in our CellCloud for enhancing trustworthiness. Since the level of trust is associated with the level of redundancy in task computation, availability of higher number of free bidders will help us in getting more precise result. In CellCloud we always have a large number of unused bidders which can be used for redundant computation.

III. Related Work

Mobile computing is used for developing several applications in the area of distributed computing. Beberg et al. proposed Folding@home, a distributed computing approach, for simulating biophysical processes [19]. According to their architecture, clients are personal computers, participate voluntarily on the computational process. The server sends the client the information of a specific work unit, which is the combination of some input files required for finishing a task within a certain time period. A header is attached inside a work unit to determine the core type while the version inside a work unit is used to download and process the core. A core can be considered as an executable file which takes input files and generates output. The output generated by the specified computational core is sent to the server. Since cores are independent from client therefore, this approach can do any type of computation and the upgrade of cores does not require reinstallation of any software. The major drawback of this approach is that once a task is given, client needs to look up manually to find and download the required core type. The client might not like the complexity associated with the operation. On the other hand, in our CellCloud architecture, once a bidder participate on bidding, all the inputs and instructions are given by the cloud. The bidder will just perform computation according to the instruction given from the cloud.

Researchers of University of Wisconsin Madison started a distributed computing project known as Condor, which manages, circulates, and upholds a wide range of computing systems [20]. Condor project allows flexibility in matching request for resource, job checkpoint and migration, and also performs remote system calls to execute jobs remotely. Dedicated resources are allocated to higher priority jobs so that, they can be finished without interruption. On the contrary, the jobs with low priority run when the CPU is idle and move to a different one when the CPU is busy. The agent, associated with Condor kernel, takes a user job and the matchmaker continuously looks for the resource which possesses the required configuration to handle that particular job. When a match is found, the job is scheduled to that specific resource. However, the primary problem in Condor project is that the low priority jobs are always interrupted if a higher priority job comes and the system does not have sufficient resources to serve the new job. Therefore, it is very hard for the client to anticipate the time to finish the job. On the contrary, in our CellCloud architecture each job is independent of each other and a dedicated amount of resources are allocated for each job. Therefore, the new job is taken only if the CellCloud system has sufficient resources to handle that job. As a result, CellCloud can provide the required time and associated cost to client before accepting any task.

Miluzzo et al. proposed the concept of mClouds where group of mobile devices, known as mDevs, are brought together to form a cloud computing platform [15]. Whenever a mobile device wishes to compute a task which requires larger resources than it currently has, the device broadcasts a solicitation message to inform the other mobile devices to join its mCloud formation. The mobile device divides and distributes the task among the mCloud devices. If the task is too large to finish even after forming the mCloud, then the device uses the backend cloud system for the remaining subtasks as no mDevs are available. In mCloud system, mobile users have to decide when to form the mCloud and whether joining on mCloud will be beneficial or not. The primary problem in mCloud system is to ensure security since anyone having a mobile device can join without verification. Therefore, the presence of a rogue device can lead to wrong output. On the other hand, all the devices used in our CellCloud are verified by the operator providing cloud service before they are included on cloud. So we can ensure more legitimate result in our CellCloud architecture than mCloud system.

Researchers from Space Science Laboratory of University of California constitute SETI@home project to explore the presence of life in the universe [21]. In their project, they consolidate immense computing power distributed all around the world to examine radio telescope signals come from space. A large number of data are broken into smaller chunks and distributed among a large number of computers for processing. Result obtained from each computer is organized by central repository. In comparison with the usual distributed systems, SETI@home uses more variety of resources distributed among diverse locations. However, the major problem in SETI@home project is that the tasks are distributed only if the resources are in idle state. Therefore, it rarely provides any real time solution for a task. Some tasks might need to be finished within a specific deadline. On the other hand in our CellCloud architecture, client can inform the deadline for their task and CellCloud selects the resources based on the client requirements.

In addition, all the aforementioned approaches have some common problems such as, motivation to participate in cloud architecture on part of the device owners is missing, the requirements of client for a task is not mentioned, and no cost benefit analysis is provided for client to take decision on joining cloud system. On the contrary, in CellCloud, the cloud service provider hires bidders by providing incentives. As a result, we can expect to get a large number of bidders for tasks. Clients can also submit their requirements for a task and the CellCloud provider provides the estimated cost of that task. Therefore, client can analyze their benefits before giving any task to CellCloud.

Moreover, we can provide chart for the client from where they can get the idea of the time and cost of their task completion. Clients can easily verify whether giving task to our mobile will be beneficial for them or not.

IV. CellCloud Architecture

The CellCloud architecture consists of a cloud central system (CCS), which is the central part of the operator cloud system. During the cloud set up, CCS sends a message to all of its base stations to inform the mobile users under their coverage area to initiate bidding. CCS determines the price for hiring a bidder based on the rating point of the bidder. The interested bidders submit the information regarding their available resources to the corresponding base stations. Each base station maintains a table consisting of the bidder id, rating point, and available resources. The high level architecture of CellCloud is shown in figure 1. However, the following two scenarios can be possible while a bidder wants to participate in a task:

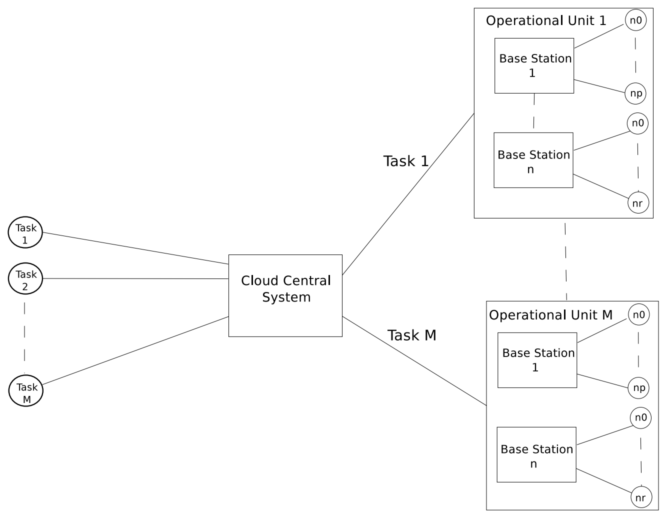

Fig. 1. CellCloud Architecture

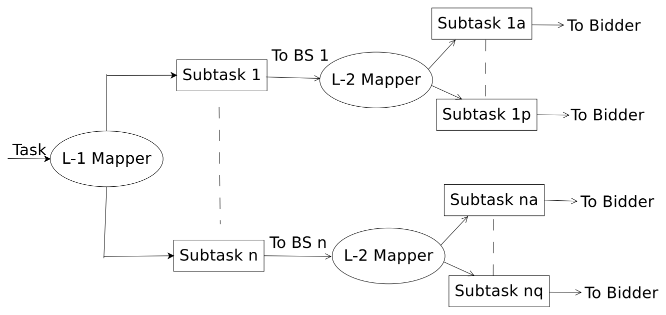

Fig. 2. Process of Mapping a Task Inside Operational Unit

Scenario 1: If the bidder is not registered and sends a request to the corresponding base station for the first time, the base station adds a new entry into its bidder information table. The base station requests CCS to assign an id for the new bidder. The id is stored into the id field of the table. A default initial rating point of 0.5 is assigned for the bidder. The bidder receives its id from the base station and stores it along with the id of the current base station.

Scenario 2: If an existing bidder requests to the corresponding base station for the participation of a task, it needs to send its id along with the id of last base station it visited for a task computation. The base station collects the information of that bidder from that last visited base station. The current base station updates its bidder table while the previous base station deletes the entry associated with that bidder.

Clients are individual users who want to take cloud facility for their tasks. Each client specifies their requirements for the task. The requirements include the task completion time along with the assurance that the task will be finished within the deadline. We refer this level of assurance as the reliability. Bidders with higher rating point ensure higher probability to finish the task within the given deadline. Before accepting any task, CCS verifies whether the CellCloud has sufficient resources to complete that task while maintaining the desired level of reliability. CCS contacts the base stations to submit the total free resources they can provide keeping the desired level of average rating point. Based on the information, CCS selects some base stations for that task. We refer each group of base stations for a specific task as an operational unit. Depending upon the free resources at each base station under the operational unit, CCS divides the task into several un-uniform subtasks. The base station with higher amount of free resources receives a larger subtask compared to a base station with lower amount of available resources. The operational unit divides the subtask again into several un-uniform subtasks and distributes among the bidders based on their resources. The process of mapping a task inside an operational unit is depicted in figure 2.

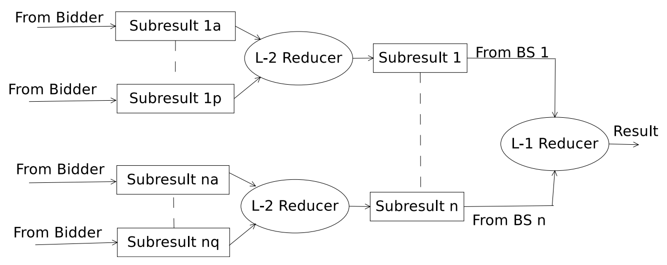

Fig. 3. Process of Reducing Result Inside Operational Unit

Each bidder sends its result to the reducer after computation. Each reducer continuously collects results from the bidders and performs reducing operation on them. Once the final result is computed by the reducer after reducing all the results obtained from the bidders, it sends the result to CCS. CCS collects the reduced result from each base station inside an operational unit and starts reducing the result. The final result is sent to the client. The process of reducing the result is shown in figure 3.

Once a subtask is finished, base station measures the performance of each bidder under its coverage area. Based on the performance, the base station re-evaluates and updates the rating point for each bidder.

V. Rating Point Calculation

The rating point denotes the level of trust the cloud provider has on a bidder of the cloud system. The rating point of each bidder ranges from 0 to 1. A bidder with a higher rating point is considered to be more trustable compared to a lower rating point bidder. Initially each bidder is assigned a rating point of 0.5. Upon the successful completion of a task, we provide a reward that will increase bidder's rating point. On the other hand, we will penalize if the bidder fails to finish the task within deadline. However, the rate of penalize will be more as compared to the rate of reward. Before sending a task to a bidder, base station will estimate the possible task completion time t_o based on its resources. In addition, base station will define two more time. The max time t_u and the min time t_l which are 10 percent larger and smaller respectively than the original estimated time. We assign a reward of 1, 0.7 and 0.5 for completing the task within the time t_u, t_o and t_l respectively.

On the other hand, a bidder will not be given any reward if it does not finish the task before t_l. The new rating point will be the average of the current rating point and the reward point. In other word, we always emphasize the most recent task during the calculation of the new rating point. Thus, if the current rating point of the users is P_c and the reward for the most recent task is P_r, then the new rating point will be,

$$P_n = \frac{(P_c+P_r)}{2}$$

VI. Measurement of Cost and TIme

For calculating the cost, we consider the cost associated with hiring the bidders and the cost associated with using network devices. The hiring time of a bidder is the addition of the time taken by the bidder to receive the task, the time bidder is used for task computation, and the time taken by the bidder to send the result. However, the consumed power in each mobile device during sending and receiving time is proportional to the rate at which the mobile device transfers and receives data. Suppose, the power drop for sending and receiving at a rate of 1 unit/s is P_x and P_y respectively. If the bidder sends and receives data at a rate of x unit/s and y unit/s then the power drop for sending and receiving is $P_x x$ and $P_y y$ respectively. If the bidder sends data for t_s times and receives data for t_r times then power consumption for receiving a task and sending result will be,

$P_{sr}=t_s P_x x + t_r P_y y$

Let us assume that the power drop during task computation is P_t in each unit of time. If the bidder takes t_t times for computing a task then the consumed power during computing a task will be,

$P_{task}=t_t P_t$

Suppose the cost for consuming each unit of power is C_u thus the cost for total power consumption will be,

$C_p=C_u(P_{sr}+P_{task})$

Let us assume that bidder with rating point 0.1 receives m% profit of total cost for power consumption. If the rating point of the bidder is R_b then the cost for hiring a bidder will be,

$C_h = \frac{(100+m)}{100} * C_p * \frac{R_b}{0.1}$

For calculating the cost associated with using network devices, we need to know the cost of mobile operator for transferring each unit of data within the network. Let us assume that the cost of mobile operator is C_{dt} for transferring each unit of data. Therefore, if the bidder receives p unit and sends q unit of data the cost of the operator will be $C_{dt}(p+q)$. If we assume that the mobile operator makes r% profit on its total cost for sending and receiving data then the cost associated with network will be,

$C_n=C_{dt}(p+q)*(100+r)/100$

Hence, the cost for completing a t unit of task by a single bidder will be,

$C_b=C_n+C_h$

If n bidders $B_1, B_2, ...B_n$ are hired for a task with the cost of $C_1, C_2, ...C_n$ respectively then the total cost for task completion will be,

$C = C_1 + C_2 + ... + C_n$

For each base station, we consider the following times during a task computation:

- Time to divide the task among n segments (t_d)
- Time to send the segments of a task to bidders (t_s)
- Maximum task completion time among all bidders (t_{max})
- Time to receive the results by base station from the bidders (t_r)
- Time to reduce the results (t_{rd})

Suppose n number of bidders $B_1, B_2, ...B_n$ participate on computation where each bidder takes x_i size of task as input and produces y_i size of result. If the data transfer rate between the bidder and the base station is BW then the time that will be taken by each base station for computing its assigned task will be,

$$t_{bs} = t_d + \frac{\sum_{i=1}^{n}(x_i+y_i)}{BW} + t_{max} + t_{rd}$$

Suppose there are n base stations $bs_1, bs_2, ..., bs_n$ inside an operational unit and each take $t_{bs1}, t_{bs2}, ...t_{bs_n}$ time to finish the task. If the CCS takes t_{div} and t_{red} time to distribute the task and reducing the result then the total time for a task computation will be,

$T = maxtime(t_{bsi}) + t_{div} + t_{red}$

Here maxtime is the maximum task completion time among all the base stations.

VII. Base Station Selection Strategy

The reliability is scaled from 0 to 1. Client submits their required reliability R along with the deadline T. CCS broadcasts a message to all the base stations and ask for total free resources it can provide with an average rating point of R. Suppose p out of n base stations reply with the resource amount $r_1, r_2, ...r_p$ respectively. Based on the task size and the deadline, CCS estimates total resources required for finishing the tasks. CCS divides and distributes the task into several smaller subtasks in such a way that each of these p base stations receives equal load on their available resources. If M is the total resources required for finishing the task of size S and CCS selects $r'_1, r'_2, ...r'_p$ resources from base stations $b_1, b_2, ...b_p$ respectively then,

$r'_i = M * \frac{r_i}{\sum_{j=1}^{p} r_j}$

VIII. Bidder Selection Strategy

Suppose a base station receives a request for resource M' from CCS with an average rating point of R. The base station uses the following procedure, ss shown in Algorithm 1, to select the bidders from its available bidder list.

Base station devides the bidders into two parts. Those bidders who have higher rating point than the target rating point R is placed into the upper part while the rest are placed into the lower part. Base station selects one bidder at a time either from upper or lower part depending upon whether the average rating point of currently selected bidders are less than or greater than the target rating point. Base station repeats

Algorithm 1 Bidder Selection

```
1: set $selectedbidders$=null
2: sort bidders into ascending order $b_1, b_2, ...b_m$ based on the rating point $p_1, p_2, ...p_m$
3: find $k$ such that $p_i >= P \; \forall i >= k$ and $p_i < P$ otherwise
4: set $upperpointer$=$k$ and $lowerpointer$=$k$-1
5: push $b_{\text{upper pointer}}$ into $selectedbidders$
6: set $upperpointer$=$upperpointer$+1
7: find $resourcesum$ which is the sum of the resources of $selectedbidders$
8: if $resourcesum \geq$M' then
9:     return $selectedbidders$
10: else
11:     if average rating point of selected bidders $\geq$ target rating point then
12:         push $b_{\text{lower pointer}}$ into $selectedbidders$
13:         set $lowerpointer$=$lowerpointer$-1
14:     else
15:         push $b_{\text{upper pointer}}$ into $selectedbidders$
16:         set $upperpointer$=$upperpointer$+1
17:     end if
18:     go to step 7
19: end if
```

the process of selecting birders until the required amount of resources for the task are achieved.

IX. Challenges in Bidding Strategy

We listed several unwanted circumstances in our proposed model which are-

A. *Finishing Task on Deadline*

One of the major challenges in our architecture is to ensure that, each task will be finished before its deadline.

In our CellCloud architecture, CCS only accepts a task if the cloud has sufficient resources. Task completion may be delayed due to some unwanted circumstances such as, network failure or bidder negligence. But bidder knows that it will receive more incentives if it has higher rating point and only finishing the task on deadline could enhance its rating point. Therefore, the bidder will sincerely try to complete the task on time.

B. *Insufficient Battery Power*

For finishing the tasks on time, mobile devices need to be remain switched on. Unexpected power failure can stop computational process or result in loss of computed data. Therefore, it is very important for CCS to know the current power status of the mobile device so that it can collect the result so far computed in case of possible power failure and distributes the remaining tasks to the back up bidders. We can use the agent SystemSens in each computational device to monitor battery power [22]. A threshold level P_{th} is defined and if the power goes below that threshold then the agent communicates with the CCS to inform possible power loss. Depending on the user's requirement, the base station may consider some of the bidders as reversed bidders and may contact them when required. Base station forwards the incomplete tasks to one of those reserved bidders on the cloud system. Base station sends a confirmation to both the current bidder and the new bidder when the incomplete tasks are transferred to the new bidder.

C. *Trusted Computing*

Another major concern for the provider is to ensure that, any malicious program inside a mobile device can not manipulate computation or compromise data inside the mobile device. Using a trusted cloud computing platform (TCCP), we can prohibit malware to access input and output data, or stop interfering during computation [23]. To establish a TCCP, a trusted virtual machine monitor (TVMM) [24] is installed in a mobile device if the platform inside the mobile device satisfies the specification defined by the trusted computing group (TCG). TVMM prohibits even the privileged user from observing or altering the data during computation. A trusted coordinator (TC) inside TCCP certifies a platform if it finds the platform secure for computation. A bidder only accepts input data and performs computation if it is certified by TC.

X. Evaluation

As part of the evaluation, we compared the time and cost of our CellCloud system with traditional cloud providers. We set up the cloud central system (CCS) and base station on our MAC book air 1.7 GHz Intel Core i5 processor with 4GB DDR3 RAM. We used Android Standard Development Kit (SDK) on Java eclipse platform for implementing and testing our CellCloud architecture. Android SDK is proprietary software of google, which is installed on the open source platform eclipse, for performing computation on android system. We used three categories of android operated mobile sets as bidder; first one is a htc mobile device with 1GHz processor along with 512 MB RAM, the second one is a motorola device with 1GHz processor along with 1GB RAM, and the third one is a nexus 4 with quad-core Krait clocked at 1.5GHz processor along with 2GB of RAM. Each mobile device has one 8GB SD card where the tasks are stored before execution. The results are also stored on the SD card. We have considered a simple word count problem as our sample task. We ran the task on the above mobile devices with different task size and measured the task completion time. We determined the average task completion time required by each of these mobile devices for a task of 1MB size and considered that task as our base task. From our sample run, we found that the task completion time on each type of mobile device is almost proportional to the task size. Therefore for simplicity, we assume that if the ratio between the current task size and base task size is S_r and the base task completion time is T_b then the current task completion time is S_rT_b. In our experimental set up, we considered that our CellCloud system can have maximum of 100 base stations where a maximum of 1000 bidders can participate under each base station. Each bidder can have any of the three types of mobile devices. The rating point of each bidder can lie between 0.1 to 1. We have also set up a private backend cloud system by hiring medium sized computer from amazon EC2 cloud on pay as you go basis which costs 12 cents/hour. For our convenience, we have referred both the hired bidders and computers as nodes.

First, we selected a task size of 1 MB. Initially we considered that the private cloud is made of only a single node

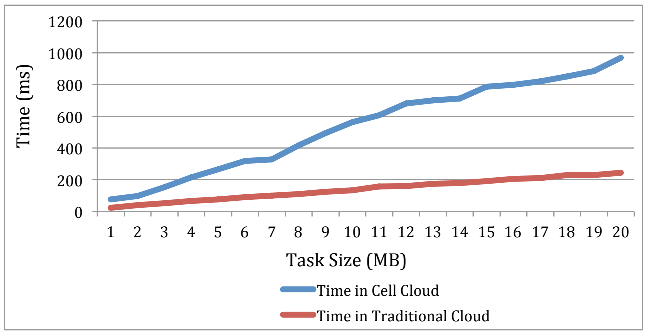

Fig. 4. Relationship of Task Completion Time With Increased Task Size for CellCloud and Traditional Cloud

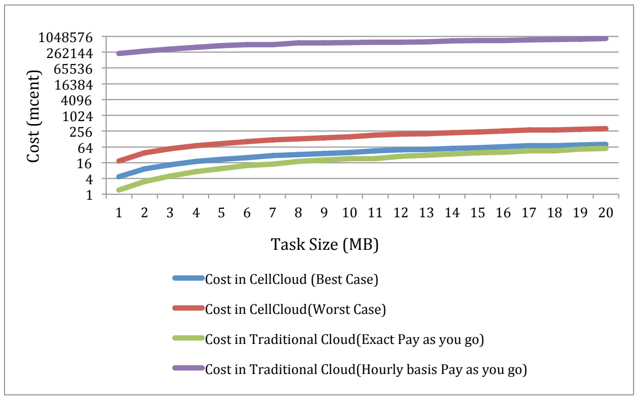

Fig. 5. Relationship of Cost with Increased Task Size for CellCloud and Traditional Cloud

and calculated the task completion time. In amazon pay as you go service, a pc has to be hired for at least an hour. Amazon charges 12 cents/hour for hiring a medium sized pc. However, in our experiment we considered two scenarios. In first scenario, we calculated the cost based on the actual time a pc is hired. Therefore, if a pc is hired for 30 minutes, the cost will be 6 cents instead of 12 cents. In our second scenario, no matter how long a pc is hired, the cost will be calculated on hourly basis. Thus, the cost of hiring a pc for 90 minutes will be 24 cents. In each iteration we added a pc until the task completion time reached the lowest value. Next we ran the same task in our CellCloud and measured the task completion time. We used the power meter to determine the power drop on a node in every ms while computing a task. We found that the power drop is approximately 1000 mw. For each node, we calculated the total power drop based on the time the node is used for computation. From the experimental result of Huang et. al. [25] we knew that, in LTE technology, the power drop for receiving data at a rate of 1 Mbps is 1340.01 mw in every ms while sending data at the same rate has power drop of 1726.43 mw/ms. In LTE, each of the 3 sectors of a base station can transfer data at a rate of 3.3 Gbit/s [26]. The bandwidth is shared among the number of nodes hired in a base station. Therefore, the nodes of different base stations might get different data rate for sending and receiving data. We calculated the power drop in every ms for this varying data rate. Based on the time a node is used for sending and receiving data, the total power drop in every node is calculated. On an average, the cost of electricity in USA is 8.75 cent/KW hour. Considering the above facts, we measured the cost of hiring bidders for the whole task. From the survey of Marshall Brain, we found that in best case, the operators can make 200% profit on their total investment if they provide internet services with just 1.9 cents per gigabyte [27] . However, in worst case they need to provide internet services with 8.3 cents per gigabyte to make 200% profit. Considering the amount of data sent and received during the mapping and reducing process, we calculated the cost in both the best and worst condition. The resulting cost is computed by adding up this network cost with the bidder hiring cost. We repeated the whole process for 10 times and took the average. Increasing file size to 1 MB in every iteration until the file sizes reached 20 MB we repeated the whole process. The time and associated costs are depicted on figure 4 and 5 respectively. We use a logarithmic Y axis of base 2 in figure 5 for our convenience.

From figure 4 we see that the task completion time in traditional cloud is almost 3 times smaller than CellCloud. From figure 5, we see that according to the first scenario of traditional cloud, the cost is little bit smaller than even the best case scenario of CellCloud. On the other hand, the cost according to the second scenario of traditional cloud is 100 times more than the cost in both the best and worst case of CellCloud. However, if we observe the pricing policy of the major cloud providers then the second scenario is more practical. Therefore, our CellCloud system might not ensure lowest time but it is far less expensive than traditional cloud.

XI. Discussion

In our simulation, we did not consider packet delay and loss during sending a task and receiving result. LTE advance used in 4G network has a very small 15 ms of latency [28]. Therefore, adding latency during simulation will slightly increase the task completion time. On the other hand, packet loss during task distribution or unexpected power failure of a bidder, might increase the time further. However, due to the availability of a large number of bidders, each bidder is expected to work for a very short period of time. Hence, the probability of power failure within this short period of time is very small. Adding redundancy during computation might stabilize the computation time but this will increase the cost. If we consider 100% redundancy in CellCloud, the cost will be approximately double of the current cost. The cost will be still much smaller than using traditional cloud on hourly basis.

XII. Conclusion and Future Work

The current trend of rapidly growing number of smart phone users along with the tendency of switching to new phones in every couple of years is creating a big pile of unused mobile devices. To the best of our knowledge, CellCloud is the first protocol that attempts to reshape the definition of mobile cloud by incorporating these unused but available resources. Along with the detailed architecture of such a system, we have developed a cost model to analyze the benefits from both mobile owners' and provider's point of view. CellCloud features, such as, facilitating different pricing options for different deadlines and level of reliability, providing money to the mobile owners for sharing their unused resources, and lessening operational cost compared to traditional cloud for the cloud provider ensure that such a model can create a win-win situation for all the parties. Currently, we are trying to build a model by which, the CellCloud provider can provide an estimation of task completion time and the associated cost before accepting a task from the client. For doing this, we are planning to train our system with sample tasks of various sizes. Based on the result obtained from the training, we will develop a map between task size and the required amount of resource. However, the actual task completion time might differ after distributing the task according to the map estimated resources. Therefore, we will include a service level agreement policy where the client will be charged based on the level of satisfaction. We also plan to deploy the architecture in small scale in real life mobile infrastructure to analyze the feasibility of the model.

References

[1] Cynthia, "The ongoing investment and hidden costs? make cloud solutions more expensive over time," http://www.contentmanagement.com/cloud-computing-myth-2, September 2011.

[2] Cade Metz, "Why some startups say the cloud is a waste of money," http://www.wired.com/2013/08/memsql-and-amazon, August 2013.

[3] D. Kawamoto, "Amazon, microsoft cloud services could suffer from shutdown," http://news.dice.com/2013/10/04/amazon-microsoft-cloud-services-could-suffer-from-shutdown-099, October 2013.

[4] 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*, pp. n/a–n/a, 2011. [Online]. Available: http://dx.doi.org/10.1002/wcm.1203

[5] M. R. Prasad, J. Gyani, and P. R. K. Murti, "Mobile cloud computing: Implications and challenges," *ISSN 2225-0506*, vol. 2, no. 7, pp. 7–15, 2012.

[6] James, "Lookout finds there are over 28 million old mobile phones going unused in the uk," http://www.tracyandmatt.co.uk/blogs/index.php/lookout-finds-there-are-over-28-million, November 2012.

[7] M. Alzain, B. Soh, and E. Pardede, "A new approach using redundancy technique to improve security in cloud computing," in *International Conference on Cyber Security, Cyber Warfare and Digital Forensic*, 2012, pp. 230–235.

[8] A. Bendahmane, M. Essaaidi, A. El Moussaoui, and A. Younes, "Result verification mechanism for mapreduce computation integrity in cloud computing," in *International Conference on Complex Systems*, 2012, pp. 1–6.

[9] H. A. Lagar-Cavilla, J. A. Whitney, A. M. Scannell, P. Patchin, S. M. Rumble, E. de Lara, M. Brudno, and M. Satyanarayanan, "Snowflock: rapid virtual machine cloning for cloud computing," in *Proceedings of the 4th ACM European conference on Computer systems*. ACM, 2009, pp. 1–12. [Online]. Available: http://doi.acm.org/10.1145/1519065.1519067

[10] J. H. Christensen, "Using restful web-services and cloud computing to create next generation mobile applications," in *Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications*. ACM, 2009, pp. 627–634. [Online]. Available: http://doi.acm.org/10.1145/1639950.1639958

[11] B. Chun and P. Maniatis, "Augmented smartphone applications through clone cloud execution," in *Proceedings of the 12th conference on Hot topics in operating systems*. USENIX Association, 2009, pp. 8–8. [Online]. Available: http://dl.acm.org/citation.cfm?id=1855568.1855576

[12] I. Giurgiu, O. Riva, D. Juric, I. Krivulev, and G. Alonso, "Calling the cloud: enabling mobile phones as interfaces to cloud applications," in *Proceedings of the ACM/IFIP/USENIX 10th international conference on Middleware*, 2009, pp. 83–102. [Online]. Available: http://dl.acm.org/citation.cfm?id=1813355.1813362

[13] X. Luo, "From augmented reality to augmented computing: A look at cloud-mobile convergence," *International Symposium on Ubiquitous Virtual Reality*, vol. 0, pp. 29–32, 2009.

[14] E. E. Marinelli, "Hyrax: cloud computing on mobile devices using mapreduce," DTIC Document, Tech. Rep., 2009.

[15] E. Miluzzo, R. Cáceres, and Y.-F. Chen, "Vision: mclouds - computing on clouds of mobile devices," in *Proceedings of the third ACM workshop on Mobile cloud computing and services*, 2012, pp. 9–14. [Online]. Available: http://doi.acm.org/10.1145/2307849.2307854

[16] A. Shye, B. Scholbrock, G. Memik, and P. A. Dinda, "Characterizing and modeling user activity on smartphones: summary," in *Proceedings of the ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, 2010, pp. 375–376. [Online]. Available: http://doi.acm.org/10.1145/1811039.1811094

[17] A. Mahesri and V. Vardhan, "Power consumption breakdown on a modern laptop," in *Power-Aware Computer Systems*. Springer Berlin Heidelberg, 2005, vol. 3471, pp. 165–180. [Online]. Available: http://dx.doi.org/10.1007/11574859_12

[18] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proceedings of the USENIX conference on USENIX annual technical conference*, 2010, pp. 21–21. [Online]. Available: http://dl.acm.org/citation.cfm?id=1855840.1855861

[19] A. Beberg, D. Ensign, G. Jayachandran, S. Khaliq, and V. Pande, "Folding@home: Lessons from eight years of volunteer distributed computing," in *IEEE International Symposium on Parallel Distributed Processing*, 2009, pp. 1–8.

[20] D. Thain, T. Tannenbaum, and M. Livny, "Distributed computing in practice: the condor experience," *Concurrency and Computation: Practice and Experience*, vol. 17, pp. 323–356, 2005. [Online]. Available: http://dx.doi.org/10.1002/cpe.938

[21] E. Korpela, D. Werthimer, D. Anderson, J. Cobb, and M. Leboisky, "Seti@home-massively distributed computing for seti," *Computing in Science Engineering*, vol. 3, pp. 78–83, 2001.

[22] H. Falaki, R. Mahajan, and D. Estrin, "Systemsens: a tool for monitoring usage in smartphone research deployments," in *Proceedings of the sixth international workshop on MobiArch*, 2011, pp. 25–30. [Online]. Available: http://doi.acm.org/10.1145/1999916.1999923

[23] N. Santos, K. P. Gummadi, and R. Rodrigues, "Towards trusted cloud computing," in *Proceedings of the 2009 conference on Hot topics in cloud computing*, 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1855533.1855536

[24] D. G. Murray, G. Milos, and S. Hand, "Improving xen security through disaggregation," in *Proceedings of the fourth ACM SIGPLAN/SIGOPS international conference on Virtual execution environments*, 2008, pp. 151–160. [Online]. Available: http://doi.acm.org/10.1145/1346256.1346278

[25] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "A close examination of performance and power characteristics of 4g lte networks," in *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '12. New York, NY, USA: ACM, 2012, pp. 225–238. [Online]. Available: http://doi.acm.org/10.1145/2307636.2307658

[26] C. Dalela, "Article: Prediction methods for long term evolution (lte) advanced network at 2.4 ghz, 2.6 ghz and 3.5 ghz," *IJCA Proceedings on National Conference on Communication Technologies & its impact on Next Generation Computing 2012*, vol. CTNGC, no. 2, pp. 9–13.

[27] Marshall Brain, "What does a gigabyte of internet service really cost? a look at the worst case scenario," http://goo.gl/FsINu, April 2011.

[28] F. Z. M. Akram, "Analysis of packet loss and latency control for robust iptv over mobile wimaxand lte assessment," *Proceedings of the International Journal of Engineering*, vol. 26, no. 3, pp. 229–240, 2013.