Advanced Adaptive Probabilities and Energy Aware Algorithm for Scheduling Tasks In MCC

Muna Eslami Jeyd
Department of Information Technology
Tehran Science and Research Branch Islamic Azad University
Tehran, Iran
muna.eslami@srbiau.ac.ir

Alireza Yari
Department of Information Technology
Iran Telecommunication Research Center
Tehran, Iran
a_yari@itrc.ac.ir

***Abstract*—Today, Mobile Cloud Computing has been widely used and can send complex computations to the stronger server with more resources and get results from them to overcome the limitations of existing mobile devices, such as battery level, the amount of CPU and memory. Local mobile clouds, which consist of the mobile devices, are used as a suitable solution to support real-time applications, especially?. Due to share bandwidth and computing resources across all mobile devices, a task scheduling is required to ensure that multiple mobile devices can effectively assign works to local mobile clouds in such way that the time limitation is considered and the amount of remaining energy is estimated for reducing energy consumption. . In this paper, we suggest energy-aware and adaptive task scheduler. The task scheduler discovers resources based on controlling messages periodically. This method, with an estimation of task completion time, calculates energy consumption and the amount of remaining energy in each processing node. Then, it schedules current work with a possible adaptive method at the processing node and sets time limitation in order to improve network efficiency under unpredictable conditions. The results of tests carried out on the proposed method compared to existing methods show that the proposed method has the lowest energy consumption per successful task. Moreover, the proposed method has scalability and high flexibility and can be deployed on any network.**

Keywords— Mobile Cloud Computing; Task Scheduling; Advance Adaptive Probabilistic Algorithm; Energy-aware;

I. INTRODUCTION

Today mobile devices are one of the most important requirements the have many lacks in calculation power, low memory, and low battery capacity. For solving this problem, mobile cloud computing is suggested. In this occasion, a mobile device connected to the remote server and result shows in mobile devices. This way is recompensed some lacks but because of delay and jitter of the network, this is inappropriate for real-time application. In general, we have to two main architecture. First one is a remote cloud, in this architecture, we have a powerful server with different hardware and software that service mobile devices through WAN network [11]. The second one is a local mobile cloud, this architecture inclusive of mobile devices that closed to each other's [12]. The profit of using load depletion is improved mobile devices energy gain through computing tasks with high energy consumption to the powerful devices without energy restrictions [12,9,13]. Task scheduling is always used for decreasing completion time or Throughput [7,10]. In general time scheduling is divided into two categories: centralized and non-centralized. In Centralized task scheduling, the decision of scheduling is doing by the central controller based on all participating node statically. Scheduling stopped if the central controller failover [8]. Due to the scalability and flexibility, non-centralized scheduling is very suitable for mobile clouds. In this approach, each node is an independent and can make decision based on their own policies and control scheduling [6]. In this way, if the decision was taken by many nodes, each node enhance itself and it causes of decrease system performance [10,11].

For solving this problem, the scheduling suggested that purpose of scheduling tasks from multiple source nodes can be transported to the nearest processing nodes. Completing tasks at the same time limit are due to the low energy consumption. But the problem of this method is not a consideration the amount of energy remaining in the processing nodes. The suggested way of solving this problem by taking the amount of energy remaining in the processing nodes. In the proposed method we have moreover the highest rate of completion work and lowest energy per work, Awareness of the amount of energy remaining in each processing node will be also.In this case, the remaining energy in a process node is determined until considered scheduling tasks in the next period.

In the following section 2, we will discuss the basic definitions and methods. in Section 3, we establish energy-aware scheduling algorithm for scheduling works. In section 4, the proposed method compared with existing methods and evaluated, And in Section 5 Conclusion of the work is done.

II. RELATED WORK

A. Mobile Cloud Computing

Mobile Cloud Computing was introduced after the concept of Cloud Computing. Mobile cloud computing, incorporating cloud computing, mobile Internet, and mobile computing respectively. Definitions of these technologies will be expressed in continue.

- Forum MCC, cloud computing will be defined as below [1]; simply mobile cloud computing, point out is an infrastructure where storage and processing of data occur outside the mobile device. Mobile apps that require more computing power and storage and run on the outside of a cell phone store in the cloud.
- MCC described Aepona [2], the method as a new algorithm for mobile applications whereby processing and data storage on mobile devices moved to a strong operating system and centralized computing in the clouds. This concentrated program is accessible with an Internet connection or a web browser on mobile devices. Mobile cloud computing can also define as a combination of the web, mobile and cloud computing definition. The most famous application access to app and services on the Internet [6,7].

B. *Mobile Cloud Architecture*

The overall architecture of mobile clouds divided into the remote cloud and local mobile cloud. In the remote cloud architecture, mobile device connects to a remote server. The disadvantage of this architecture is its high latency of WAN network. Therefore it's inappropriate for real-time applications. The second type is local mobile cloud architecture. The mobile architecture itself as a provider of cloud-based resources. This network architecture is used in mobile ad hoc network (MANET). In this architecture works are sent to the nearest processing node, to reduce latency in the process of offloading. Therefore second architectural, mobile cloud architecture is suitable for real-time applications. In [8] and [11] has been shown the effectiveness of using the mobile cloud computing in discharge locally on schedule. The cloudlet architecture is based on the second architecture [11]. Cloudlet architecture suggests a middleware framework support real-time applications. In this context, calculation of discharge occurs again, when a defect is detected or functional limitations occur. The focus of the proposed method is the utilize discharge to the local mobile cloud is similar to the past works. But the focus of this method is to create an independent work with the ability to deliver.

C. *Scheduler Architecture*

Mobile Scheduling architecture plays important role in the scalability, independence and performance of the system. Schedule objectives in a local mobile cloud environment usually related to work completion time and energy consumption.

The proposed method has been used non-centralized architecture for scheduling. related work Studies showed that there are different criteria for the delivery of computing servers. The following equation is a fundamental condition for improving the work completion [9,12].

$$\frac{w}{s_m} > \frac{d_i}{B} + \frac{w}{s_s} \qquad (1)$$

OnSuppose w is the amount of computation and di is the size of data needs to be transmitted for remote execution. Let S_m be the speed of the mobile device and S_s be the speed of the server. B is bandwidth between the mobile device and the server. On the left side of the equation, needed time to carry out the task on mobile devices, on the right side, the time required to perform a task on the server with the transmission time. offloading can expedite the completion of the work to improve performance [7]. In [6], was introduced scheduling in a dynamic and non-centralized approach called Community aware scheduling algorithm (CASA) for the mobile network. The algorithm includes transmission phase and dynamic scheduling phase, both phase to ensure rapid deployment and optimization work together. In this method, the job response time shows the time between arrival and the time get a result of the job. If all nodes sent their demand to the processing node, jobs will be done in the shortest response time. It also reduces average waiting time in the queue. In [18], A plan based routing QoS OLSR as proposed. This procedure uses real-time application with heavy computer benefit for transmission. It shows cloud service in MANET network [16,17]. In this way, the control messages are broadcast periodically on the network. Context on each node after receiving routing messages are updated. Update the information frequently give rise to a burden on the network, which can increase the number and size of routing packets. Due to problems with previous methods, the proposed methods in addition to solving problems, as well as plans to calculate the amount of energy consumed Finishing time in each processing node, the processor calculates the amount of residual energy at each node and use it to reschedule tasks. This method of non-centralized scheduling is based on local cloud mobile architecture this feature has caused the flexibility, scalability, and enhanced functionality. is one of the most important requirements the have many lacks in calculation power, low memory, and low battery capacity. For solving this problem, mobile cloud computing is suggested. In this occasion, a mobile device connected to the remote server and result shows in mobile devices. This way is recompensed some lacks but because of delay and jitter of the network, this is inappropriate for real-time application.

III. THE PROPOSED ALGORITHM

A MANET networks consist of a number of wireless nodes that can be an undirected graph G (V, E) is modeled. If two nodes consider the following: $u\epsilon V, v\epsilon V$. We have one mane$(u,v) \in E$ if and only if $dis(u,v) \leq rt$, where dis(u, v) is Euclidean distance between node u and v.

A direct connection between two nodes, which the distance between them should be within radio range rt, Computational processing power of node u is shown by mips (million instruction per second). According to [14], Eqn. (2) gives that the power e consumed by a CMOS processor, in watts, is equal to the activity factor α of the system (percentage of gates that switch for each cycle, on average 50%) multiplied by the capacitance C of the CPU, the voltage V squared, the frequency f.

$$e = \alpha C V^2 f$$

However, obtained data shows that the power of modern CPUs is in line with the second power of work cycle. So in this article square of that was applied [15], Eqn. (3). Suppose that the energy consumption of processors in each instruction in line with mips_u then:

$$eu \propto mips_u^2 \quad (3)$$

In each node $t_{q,u}$ delay queue that refers to the waiting time in the queue until the moment the work is labor-processor systems. Energy consumption execution of the work consists of two parts:

- Computational energy: energy that is wasted for execution of the work by the processing node u.

$$Computation\ Energyj = e_u \times C_j \quad (4)$$

- Communication energy: consumed energy within processing node for unloading during communication. Community energy consumption proportional to the amount transferred or received data. H (j) is the number of hops from the source node to processing node.

$$Computation\ Energyj = 2 \times H(j) \times (e_t + e_r) \times D_j \quad (5)$$

The total energy consumed by the task j is as follows:

$$Task\ Energy_j = Computation\ Energy_j + Communication\ Energy_j \quad (6)$$

The total energy consumed by task n is as follow:

$$Total\ Energy = \sum_{i=1}^{n} Task\ Energy_j \quad (7)$$

Completion time of task consist of these three parts:

- $t_{q,u}$: Queuing time processing node u where the is time to wait before running the task.
- $t_{exection}$: execution time in processing node u

$$t_{execution} = C_j / mips_u \quad (8)$$

- $t_{transmission}$: The time required to transfer data between source and destination nodes.

$$t_{transmition} = 2 \times (H(j) - 1) \times D_j / B_{u,v} \quad (9)$$

Completion task j, time given to the node u is, consist of three general parts:

$$Task\ completion\ time_j = t_{q,u} + t_{execution} + t_{transmition} \quad (10)$$

When a task has a good scheduler, if it is completed before the deadline. That met through Eqn. (11). Otherwise, it fails.

$$Task\ completion\ time_j < T_j \quad (11)$$

n is a number of all tasks, for scheduling task we have to criterion:

(2)

$$Completion\ Rate = Number\ of\ Successful\ Task / n \quad (12)$$

$$Average\ Energy\ per\ Successful\ Task = \frac{Total\ Energy}{Number\ of\ Successful\ Task} \quad (13)$$

According to the lack of previous methods that were discussed in previous sections, we have proposed an advanced adaptive probabilistic scheduling for task scheduling. Where consist of two phase: advance resource discovery phase, advanced adaptive probabilistic scheduling phase. In the first phase, the source node is able to get information about the context of adjacent processing nodes. In the second phase, a processing node selected for task j. The performance of the two separate phases, improving the efficiency of local clouds.

Advance resource discovery phase: the proposed method is based on the QoSOLSR. We recommend that the parameters of the Routing table by adding the following parameters of neighboring nodes to change two HELLO messages and TC message.

- Device parameter: mips_u processing speed and e_u power consumption factor, re_u remaining energy in the node u.
- Queue length: $t_{q,u}$ is the current queue length in the node u.

In this phase, using the following message at different time intervals message TC, discover and detect contextual information about each node.

```
Resource Discovery Algorithm
  Input : Control message (Hello message or TC message )
  Output : Neighbor table (The table storing the node parameter for each node)
  Function : Handle control message and update the neighbor table at node u
  Procedure body :
  {
  Initialize  the neighbor table
  Listen control message
  If (Message Type == HELLO_MESSAGE)
  {
          Update the neighbor table
          If (nod u is an MPR node)
               Construct/ update TC message
  }
  If (Message Type == TC_MESSAGE)
  {
          Update the neighbor table $t_{margin} = t_{margin} + \Delta t_1$;
          If (nod u is an MPR node)
               Forward the TC message to all of its neighbors
  }
  }            $t_{margin} = t_{margin} - \Delta t_2$;
```

Fig. 1. Resource discovery algorithm

Advanced adaptive probabilistic scheduling phase: After the discovery of nodes and their information, whenever a source node u get a task j that provided by your local user. Estimated expenditure energy of task $Energy_{j,v}$ and completion time of task j, the method of the processing node $time_{j,v}, V \epsilon P$. These two variables are obtained according to Eqn 6 and 10. Scheduler 'p' kept series of processing nodes that estimated out of the Eqn 14. The scheduler selects a node, according to the amount of energy remaining in each processing node, a node selects. The probability $V \epsilon P$ of a processing node is selected p_v.

$$Task\ completion\ time_{j,v} < T_j - t_{margin} \tag{14}$$

$$\rho_v = \frac{1}{Task\ Energy_{j,v}} / \sum_{v \epsilon C} \frac{1}{Task\ Energy_{j,v}} \tag{15}$$

$$Energy_{remaining} = Energy_v - Energy_j \tag{16}$$

In the proposed algorithm, a1 is failed the task, Δt_1 is the time difference, a2 is a successful task, Δt_2 is a time difference of scheduler.

```
Advance adaptively probabilistic scheduling algorithm
Input: (1) Neighbor table from phase1, (2) task set J
Output: scheduling (mapping)
Function: schedule the task to processing node in local mobile cloud
Procedure body:
Set the number of consecutive successful tasks current $a_2$=0
Set the number of consecutive failed tasks current $a_1$=0
Set the number of consecutive successful tasks $n_s$ = 0
Set the number of consecutive failed tasks $n_f$ = 0
Task j ε J arrives at source node u, record current time $t\_start_j$.
$P` \leftarrow \emptyset$
{     Estimated task completion $time_{j,v} = t_{q,v} + t_{execution} + t_{transmition}$
              Task Energy$_{j,v}$ = Computation Energy$_{j,v}$ + Communication Energy$_{j,v}$
           $Energy_{remaining} = Energy_v - Energy_{j,v}$
           If (estimated task completion $time_{j,v} < T_j - t_{margin}$ && $Energy_{remaining} \geq 0$)
                         $P` = P` \cup v$}
      {
           $\rho_v = \frac{\frac{1}{Task\ Energy_{j,v}}}{\sum_{v \epsilon P`} \frac{1}{Task\ Energy_{j,v}}}$
      }
      $t_{q,v} = t_{q,v} + \frac{c_j}{mips_v}$
      Task completion $time_j = t\_complete_j - t\_start_j$
      If (Task completion $time_j \leq T_j$) {
            $n_s$ ++;
           $n_f$ = 0;}
      else {
         $n_f$++;
         $n_s$ = 0;}
      If ($n_f > a_1$) {

      $Energy_v = Energy_{remaining}$;
      $a_1 = n_f$;}
      If ($n_s > a_2$) {

      $Energy_v = Energy_{remaining}$;
       $a_2 = n_s$;}
```

Fig. 2. Advanced adaptive probabilistic algorithm

IV. THE PROPOSED ALGORITHM

To evaluate the performance of different scheduler we used in the Local mobile cloud with Omnet ++ simulator. In this simulation, the local mobile cloud consists of a group of nodes, each node capable of transmitting a radio signal to about 40 meters through a wireless cable with bandwidth about 11Mbps and 802.11g standard. In this simulation scenarios, we have different tracks with 10 and 20 nodes, the source node and 16-node, the processing node. In order to optimize the energy consumption of task time and energy consumption for each processing node estimated and calculated.

A. The Impact of different data size

Is We changes the volume of data in a fixed amount for calculations. Totally we have three type of data volume: small, medium and large. I small volume of data, size is uniform [1500,500] bytes and distributed in the network. In cases where the volume of data is medium and large, the uniform calculation is [4500.3500] byte and [8500.7500] byte.

Table 1. Three different data size scenarios

	Average data size (byte)	Average computation amount (million Instructions)
Small Data	1000	100
Medium Data	4000	100
High Data	8000	100

Figure 3 shows that when the volume of data increases, we saw a sharp drop in the rate of completion of the task. All five scheduler, in this case, are the same. The reason for this problem is that with the increasing volume of data, tasks needed more time for data transmission. Because the transmission media is shared between nodes and resulting in increasing waiting time. When volume changes are medium or high, the rate of completion task on the advanced adaptive scheduling decreased only 9.7%. In schedulers, probabilistic, adaptive probabilistic, advanced adaptive probabilistic and greedy task completion rate respectively %12.1, %16.5, %12.1, and %9.8.

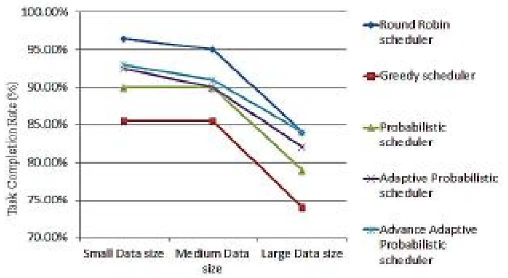

Fig. 3. Task completion rate vs. different data size

Figure 4 shows that the increasing volume of average consumption energy for each successful task in all five methods work successfully. This would reduce the rate of completion of the task and increased communication energy. By changing the volume of data from low to high, average power consumption for each successful task in shifts circulation scheduling increased 13.87%. In schedulers, probabilistic, adaptive probabilistic, advanced adaptive probabilistic and greedy task completion rate respectively %20.53, %23.0, %25.66 and %25.67. Advanced Adaptive probabilistic Scheduling likely to have the greatest increase in energy, Because this tends to choose more powerful processing node, failed to improve the rate of completion of the task at a time when tasks, There are shown in Figure 3.

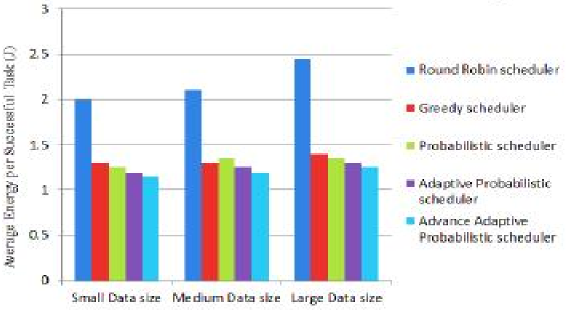

Fig. 4. Average energy per successful task vs. different data size

B. The Impact of different Scale of Network

Network size also affects the scheduler performance. In this experiment, the timeframe 0.4 seconds is considered for TC message. Figure 5 and Figure 6 shows that in a large network has lower completion rate, but more energy can be saved. In a large-scale network in all five of scheduling, the completion rate was reduced about %5 percent. As a result, scheduling occurs harder. And in some cases can reduce the performance of the system, This resulted in reducing completion of task rates to 8.73% in the greedy scheduling. While the Advance Adaptive Probabilistic Algorithm in both network scale is only about %5 percent decline.

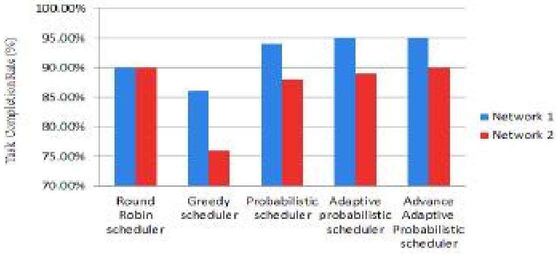

Fig. 5. Task completion rate in small and large network

In figure 6, all five scheduling have a %12.5 higher energy consumption than the large-scale network. The main reason is energy consumption for transmission and reduces the rate of task completion. Compared with other scheduling Advanced Scheduler has average energy potential comparative in large networks to 29.5% and in small networks is reduced to 11%.

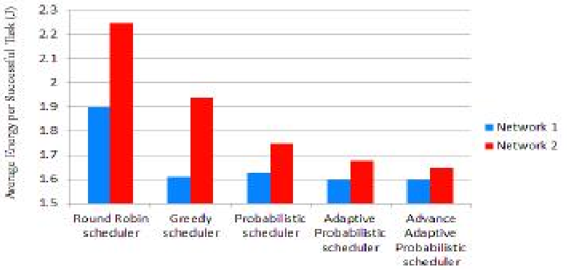

Fig. 6. Average energy per successful task in small and large network

V. CONCLUSTION

One of the most important challenges in the field of mobile cloud computing systems reduces energy consumption and Shortage computing power. Hence we need Assigned task with complexity and requirement to more energy to the cloud computing. Therefore, there must be scheduled to send these items to the cloud. In this paper, we propose a probabilistic algorithm for the mobile cloud, especially in real-time application. The proposed scheduling is based on QosOLSR. In this type of scheduling, source node receives context via a periodical control message. And discover neighbor node via source node. In this suggested method, in order to allocate for increasing efficiency and system performance, the amount of energy in each processing node was calculated. It means that two parameter t_{margin}, $Energy_{remaining}$ is set in a way to improve efficiency in unpredictable condition, and the allocation will act according to these two parameters. In general, the experimental results shows that the Advance Adaptive Probabilistic scheduling might be able to reduce the average energy consumption per successful tasks. Whereas the completion rate is also maintained. Scheduling also can be provided for fixed and mobile networks and is compatible with different types, this type of scheduling is suitable for real-time tasks. Because the local mobile cloud architecture is scalable and flexible.

used are used. In this protocol, both nodes trust and confidence are used. The trust by each node for each routing is calculated in its routing table and the levels of trust depend that a packet can reach its destination. Trusted node confidence values are calculated based on the difference between the advertised destination node and confidence started to transfer current. Then, these values are used in the routing decision.

References

[1] Raffaele Bolla, Roberto Bruschi, Franco Davoli, (2015), "Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures", IEEE Communications Surveys & Tutorials.

[2] Behrang Barekatain, Shahrzad Dehghani , Mohammad Pourzaferani, (2015), " An Energy-Aware Routing Protocol for Wireless Sensor Networks Based on New Combination of Genetic Algorithm & k-means", The Third Information Systems International Conference.

[3] Fadoua Yakine, Abdellah Idrissi, (2015), "Energy-Aware Topology Control And QoS Routing In Ad-Hoc Networks", The 10th International Conference on Future Networks and Communications.

[4] Keke Gai, Meikang Qiu, Hui Zhao, Lixin Tao, Ziliang Zong, (2015), "Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing", Journal of Network and Computer Applications, Volume 59, pp. 46-54 , 11 June.

[5] L. Shakkeera, Latha Tamilselvan, (2015), "Energy-Aware Application Scheduling and Consolidation in Mobile Cloud Computing with offLoading", Emerging Research in Computing, Information, Communication and Applications, pp. 253-264.

[6] Ziqian Dong, Ning Liu, Roberto Rojas- Cessa, (2015), " Greedy scheduling of tasks with time constraints for energy-efficient cloud computing data centers", Journal of Cloud Computing: Advances, Systems, and Applications.

[7] Y. Wang, Chen, I. Wang, R. Wang, D. Wang, C. Wang, (2014), "A survey of mobile cloud computing applications: perspectives and challenges", Wireless Pers.Commun.

[8] X.Lin, Y.Wang, Q.Xie, M.Pedram, (2014), "Energy and performance-aware task scheduling in a mobile cloud computing environment", IEEE International Conference on Cloud Computing.

[9] Meikang Qiu, Zhi Chen, Zhong Ming, (2014), "Energy-Aware Data Allocation With Hybrid Memory for Mobile Cloud Systems", IEEE SYSTEMS JOURNAL.

[10] CHathura M. Sarathchandra Magurawalage, Kun Yang, Liang Hu, Jianming Zhang, (2014),"Energy-efficient and network-aware offloading algorithm for mobile cloud computing", Computer Networks 74 ,22–33.

[11] Z.Sanaei, S.Abolfazli, A. Gani, R.Buyya, (2014),"Heterogeneity in Mobile Cloud Computing: Taxonomy and Open Challenges", IEEE COMMUNICATIONS SURVEYS & TUTORIALS, vol. 16, NO.1.

[12] Niroshinie Fernando, Seng W.Loke, Wenny Rahayu,(2013),"Mobile cloud computing: A survey", Future Generation Computer Systems 29, pp. 84-106.

[13] Hong T.Dinh, Chonho Lee, Dusit Niyato & Ping Wang,(2013), "A survey of mobile cloud computing: architecture, applications, and approaches", Wirel. Commun. Mob. Comput.:pp.1587–1611.

[14] Suraj Sindia et al.,(2013), "MobSched: Customizable Scheduler for Mobile Cloud Computing," in 45th Southeastern Symposium on System Theory, pp. 129-134.

[15] Karthik Kumar, Jibang Liu, Yung-Hsiang Lu, and Bharat Bhargava,(2013),"A Survey of Computation Offloading for Mobile Systems," in Mobile Networks and Applications, pp. 129-140.

[16] Tim Verbelen, Pieter Simoens, De Turck Filip, and Dhoedt Bart, (2012), "Cloudlets: Bringing theCloud to the Mobile User," in Proceedings of the third ACM workshop on Mobile cloud computing and services, pp. 29-36.

[17] Asaf Cidon, Tomer M London, Sachin Katti, Christos Kozyrakis, and Mendel Rosenblum,(2011),"MARS: adaptive remote execution for multi-threaded mobile devices," in MobiHeld '11 Proceedings of the 3rd ACM SOSP Workshop on Networking, Systems, and Applications on Mobile Handhelds, Article No. 1.

[18] Shu Shi, Cheng-Hsin Hsu, Klara Nahrstedt, and Roy Campbell,(2011),"Using graphics rendering contexts to enhance the real-time video coding for mobile cloud gaming," in MM '11 Proceedings of the 19th ACM international conference on Multimedia, pp. 103-112.

[19] Yamini Nimmagadda, Karthik Kumar, Yung-Hsiang Lu, and C. S. George Lee, (2010),"Real-time Moving Object Recognition and Tracking Using Computation," in The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2449-2455.

[20] Romano Fantacci, Daniele Tarchi, and Andrea Tassi,(2010),"A novel routing algorithm for mobile pervasive computing," in Global Telecommunication Conferrence, pp. 1-5.

[21] Gartner: the world's leading information technology research and advisory company. [Online]. http://www.gartner.com/.

[22] Eduardo Cuervo et al.,(2010)," MAUI: making smartphones last longer with code offload," in Proceedings of the 8th international conference on Mobile systems, applications, and services, pp. 49-62.

[23] Roelof Kemp et al.,(2009),"eyeDentify: Multimedia Cyber Foraging from a Smartphone," in ISM '09 Proceedings of the 2009 11th IEEE International Symposium on Multimedia, pp. 392-399.

[24] Ahmad Afsahi Ryan E.Grant,(2006),"Power-Performance Efficiency of Asymmetric Multiprocessors for Multi-threaded Scientific Applications," in IPDPS'06 Proceedings of the 20th international conference on Parallel and distributed processing, pp. 25-29.

[25] Murali Annavaram, Ed Grochowski, and John Shen,(2005), "Mitigating Amdahl's Law through EPI Throttling," in Proceedings of the 32nd annual international symposium on Computer Architecture, pp. 298-309.

[26] Omnet++ community site. [Online]. http://www.omnetpp.org/.

[27] Amazon elastic compute cloud (EC2). [Online]. http://aws.amazon.com/ec2/.

[28] Xiaoshuang Lu, Hossam Hassanein, and Selim Akl, ,(2005),"Energy-aware dynamic task allocation in mobile ad hoc networks", International Conference on Wireless Networks Communications and Mobile Computing, pp. 534-539.