

Cloud- Based Mobile Augmentation in Mobile Cloud Computing

Alshamaileh Mohammad^{1, a} and Li Chunlin^{2, b}

^{1,2}*School of Computer Science, Wuhan University of Technology, Wuhan, CHINA*
^a*moh353_moh@hotmail.com*, ^b*moh353@gmail.com*

Abstract

Mobile devices are increasingly becoming famous in the current times. Limited computing resources are still a hindrance for resource intensive computations. Several frameworks have been established to enable cloud -based mobile augmentation (CMA) approach be followed by mobile applications. CMA has a strategy of dynamic outsourcing resource intensive tasks to the external resources. CMA's state of art mobile augmentation model employs resource- rich clouds to increase, enhance and optimize mobile device. Augmented mobile devices in the cloud are geared towards extensive computations and storage of big data beyond the device capabilities with little or no traceable footprint and vulnerabilities. CMA has made efforts towards exploiting various cloud- based computing resources especially the distant clouds and proximate mobile nodes to augment mobile devices. Task scheduling is one of the approaches that are well elaborated using computations in this paper that is employed to reduce execution time.

Keywords: *Cloud computing, Cloud-based Mobile Augmentation, Mobile Cloud Computing, Task Scheduling*

1. Introduction

Cloud computing is an emerging concept that entails a combination of many fields in computing. Mobile cloud computing is a combination of cloud computing tools with the mobile devices making them resource full in terms of computational power, storage and energy [1]. It is a new paradigm that is slowly but surely gaining popularity. Cloud computing is founded on service delivery, increasing storage, system automation, accessibility of software over the internet, flexibility and ease of information mobility [1]. Use of mobile devices has immensely increased over time and its utilization on the cloud is now a reality. Availability of powerful mobile end-user devices and the continuous wireless network improvement has enhanced mobile computing making it a predominant computing paradigm [2].

Cloud-based mobile augmentation (CMA) is a computing concept that allows resource-constrained devices to offload resource intensive tasks to other device or cloud computing resources [2]. Mobile cloud computing creates a need for specialized application development models that support computational offloading from mobile devices to the cloud. A successful exploitation of cloud resources in other stationary devices has motivated the utilization of elastic resources for the mobile devices in the wireless ecosystem resulting into the emergence of mobile cloud computing as new computing field [3].

Cloud mobile application seeks to minimize mobile gadgets' resource consumption through provisioning rich cloud resources without any degradation [3]. Portability has been appreciated by end users leading to an incredible increase in mobile devices sales especially Smartphone greatly surpassing the desktop sales. The conventional applications on the sedentary desktops should be redesigned to operate on a mobile platform [4]. Limited power storage of mobile devices especially Smartphone was found to be a problem due to reduced operational time. The study in 2005 involving 15 countries shows

that extended battery time on mobile devices is one of the most important features compared to storage and cameras [5].

Cloud computing is the distributed computing model that provides computing services and resources on demand basis by implementing pay-as-you-go, on-demand provisioning and utility computing business model [13]. Emergences of cloud computing enable mobile device users to access shared computing resources. The constraining of processing powers, data synchronization, battery life, storage reliability and scalability of mobile devices has led to the mobile cloud computing (MCC) paradigm being developed. While some mobile applications rely on the resources and capabilities of the mobile device only, others require constant connection to provision distinct cloud resources for their full functionality [14].

Feature offloading which is common practice in MCC uses remote method invocation to perform computational offloading with surrogates. It is beneficial to this paradigm since it integrates well with the principle of object oriented design but requires constant synchronization of shared variables. Cloud resources have created an opportunity to mitigate the shortcomings of utilizing powerful computing in augmenting mobile devices such as Smartphone, tablet PCs and PDAs among others [15].

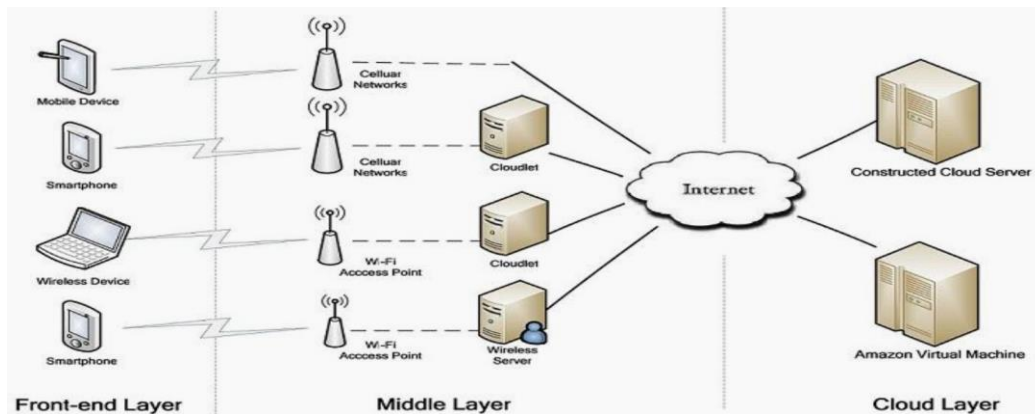


Figure 1. Cloudlet Architecture

Main requirements of MCC include availability, portability, scalability, usability, maintainability and security. In order to achieve availability, reliable network connection is required for computation offloading. The challenge that exists currently is that of intermittent connectivity in wireless connections. Regardless of increase in the use of mobile devices, there are several inherent shortages that continue to impede feasibility of intensive mobile computing operations [15]. There are several issues that necessitate cloud-based mobile augmentation to meet the increasing needs of mobile device users as shown in the figure below:

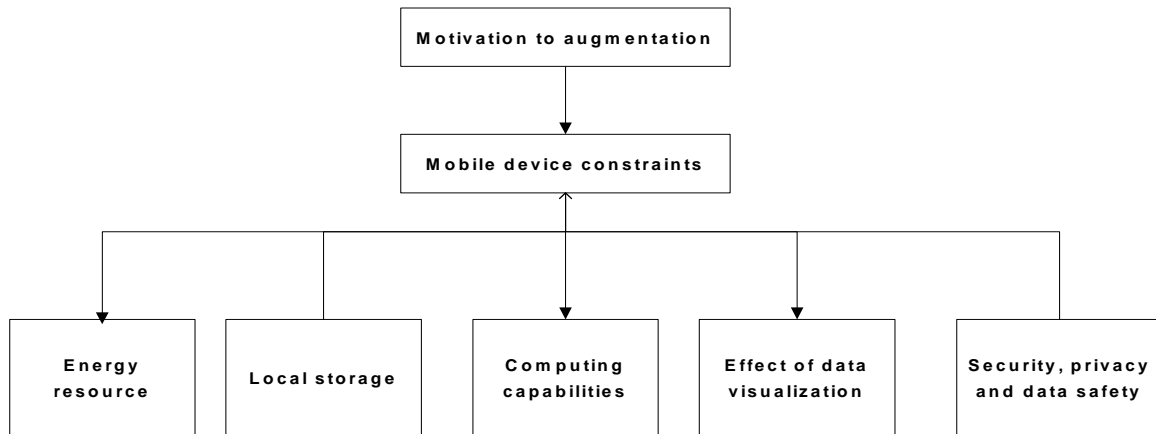


Figure 2. Mobile Issues that Motivate Augmentation

1.2. Cloud-based Mobile Computation Augmentation

The need to empower computation capabilities of mobile devices is not a new concept and there have been several approaches towards achieving this goal. The approaches include load sharing, remote execution, cyber foraging and computation offloading [3]. Remote execution is a concept that emerged in the 90s with the researchers aiming at enabling mobile computers performs remote computations and data storage to conserve their scarce native resources and battery life extended. Cyber foraging introduced with the concept of surrogate computing by Satyanarayana (2001). This concept is defined as a process to dynamically augmenting the computing resources of a wireless mobile computer by exploiting wired hardware infrastructure.

The augmentation of the resource constrained mobile devices is achievable through migration of the entire or partial application resources to cloud -based computing resources. These resources include the remote cloud data centres [20], group of nearby mobile computing devices [21], hybrid model with heterogeneous computing resources [22] or to the proximate resourceful computing devices [23].

2. Related Work

CMA is a computing paradigm that seeks to utilize the cloud for resource constrained devices. Surrogate computing concept introduced in 2001 by Satyanarayanan [6] is closely related to CMA. Cyber foraging is one of the solutions for enhanced mobile application performance by which users can perform computation and data processing on nearby fixed computers [6]. Offloading tasks that involve heavy computational work is one requirement for mobile users. Fernando, Loke & Rahayu [7] introduces an approach that does not require building an infrastructure to offload heavy computational work. Instead of using cloud services such as Amazon EC2, a local cloud is created using nearby mobile devices [7]. Use of nearby mobile devices to offload heavy computation load is a cheaper method compared to building of infrastructures. The model of cyber foraging band surrogate computing is faced with tough challenges that makes it less applicable as compared to cloud- based mobile augmentation in mobile cloud computing. The security support aspect required making surrogates minimally intrusive for the data and user safety still haunts. The advance time notice for effective server staging, proper load balancing and improved trust levels in the surrogates are required for this method to be actualized. The cloudlet model for cloud- based mobile augmentation in mobile cloud computing gives better solution to the challenges facing surrogates. The security levels are highly rated when using the cloud- based mobile augmentation in mobile cloud

computing. The model is easier and convenient to implement since there are dedicated mobile devices that handle the load as opposed to cyber foraging band surrogate computing which is dependent on the availability of nearby mobile devices.

Cloud operating system (COS) is a software structure which can support autonomous workload elasticity and scalability based on the application level migration strategy for reconfiguration [8]. Tolia, Andersen & Satyanarayana [9] concludes that long WAN latency decreases application responsiveness by investigating application interactivity when mobile devices are augmented using remote servers. Use of this paradigm would greatly influence workload offloading elasticity and scalability. Use of the Cloud operating system would greatly scale the workload up by migrating the mobile actors and terminating idle virtual machines. Scaling up workload offload would thereby improve the performance of the entire cloud. Much focus has been laid on energy saving, performance improvement or both, and minimum bandwidth cost. The subscriber mobility when utilizing remote servers for augmentation and factors such as weather and distance makes bandwidth variations high leading to unreliability and increased time loss. Unreliability is undesirable when moving towards better performance satisfaction, energy saving and minimization of bandwidth costs. The model presented in this paper takes much consideration on minimizing the augmentation cost together with better reliable offloading techniques through use of cloudlets.

Adoption of profiling- partitioning technique to identify offloaded parts of a program for energy saving, performance improvement has been dedicated to offload data or part of the application to remote servers to reduce execution time [10]. Cost reduction in terms of bandwidth cost and the impact of it on energy consumption for mobile device in the cloud- based mobile augmentation environment in MCC domain [10]. The aim is to reduce the overall cost of improved uptake of augmented resources on the cloud. The cost related challenges are well mitigated through the profiling-partition techniques due to easy offloading of programs in parts with ease of identifying the offloaded parts thereby reducing execution time. Offloading of resource intensive workload to remote servers is time consuming and expensive due to the distance and workload being processed by the remote servers. The urgent need for nearby resources offloading for a temporary execution of tasks is highly constrained when the remote servers are utilized. The absence of task scheduling scheme in this methodology contributes to delays and inefficiencies during resource uploading. The use of cloudlet paradigm and application of efficient task scheduling algorithms as presented in the paper are more desirable as opposed to the method presented by Nyamajejen and Yu [10]. The distance is greatly reduced since cloudlets are strategically positioned and the tasks are scheduled in such a way that collisions are reduced during offloading.

Cloud is equated to a distributed system. Job allocation algorithms in the distributed systems can be classified as static or dynamic [11]. [12] Introduces a job scheduling algorithm for optimal heterogeneous tasks scheduling on heterogeneous sources according to the Genetic Algorithm that reach to better efficiency and reduced delay times. The job scheduling algorithm proposed in this model can be more applicable in dynamic job allocation while it is constrained in the static job allocations due to the focus only on heterogeneous tasks in heterogeneous sources. The distributed nature of the algorithms proposed makes the model more prone to sophisticated distributed attacks such as the distributed denial of service and cross site scripting which can be fatal if successfully done. Use of cloudlets as presented in this paper improves the security during offloading of tasks into the augmented mobile cloud environment. The algorithms presented in this paper provide an efficient job scheduling so as to enhance the speed and accuracy of offloading individual tasks on the cloud.

3. Problem Statement

Mobile cloud computing is a new paradigm in the computing technology. Time spent when a mobile device wants to augment resources to the cloud and utilize them is of great importance. As it stands today time spent to upload application or resources to the cloud and also time required to access resources from the cloud is large. In view of execution time is affected by several factors as shown below.

3.1. Execution Time

The total time consumed to execute a transaction in a cloud based mobile augmentation in the cloud computing can be calculated as follows:

$$T_{total} = T_{mobile} + T_{RT} + T_{cloud} \quad (1)$$

Where T_{total} is the total time consumed in transaction execution while T_{mobile} , T_{RT} and T_{cloud} represents time for native computation, roundtrip delay and computing latency of the cloud respectively.

The computation latency is dependent on the CPU clock speed, RAM size and I/O performance whereas roundtrip delay includes transmission, propagation, processing and queuing delay.

$$T_{RT} = 2 * (T_{Prog} + T_{trans} + T_{proc} + T_{que}) \quad (2)$$

T_{RT} is the round trip delay

T_{prog} represents propagation delay which represents the time needed to transmit the packet through a medium from client to server.

$$T_{prog} = 4 * H * (D/S) \quad (3)$$

T_{proc} is the processing delay time which represents the time an intermediate node can handle the packets on the network.

$$T_{proc} = 4 * (H-1) * T \quad (4)$$

T_{tran} is transmission delay time that depends on the amount of data being transferred over the network bandwidth β .

$$T_{tran} = P * (PS / \beta) \quad (5)$$

T_{que} is the queuing delay which represents the time packet queues before transmission.

$$T_{que} = (H-1) * (PS / \beta) \quad (6)$$

Substituting 3, 4, 5 and 6 in 2

$$T_{RT} = 2 * (4 * H * D/S) + (P * PS / \beta) + ((H-1) * PS / \beta) + (4 * (H-1) * T_{PROC}) \quad (7)$$

This shows that the roundtrip latency highly depends on the number of hops between mobile devices and cloud rather than the distance between the cloud and the communicating mobile device. The scheduling of tasks in the cloud is of major importance.

4. Proposed Solution

Due to many delays as indicated in the problem statement, a model that seeks to schedule tasks is proposed to reduce execution time spent by each node (mobile device) to access or upload resources to the cloud.

The user according to Poisson process generates a task request to the scheduler with the user k able to generate jobs with an average rate of β_k (jobs per second). Scheduler receives tasks from users and dispatches each to computing nodes assuming that the cost of task decomposition is negligible.

According to the number of computing nodes available, scheduler i moves users tasks into m task slices, a_{ij} is the ratio that scheduler i uses to assign a task to computing node j satisfying the constraint that $a_{ij} \geq 0$ & $\sum_{j=1}^m a_{ij} = 1$

The computing node executes and processes the tasks with the assumption that the computing node has the ability to execute the general task slice. The constraints below are meant to enhance stability of the scheduler.

$$\sum_{i=1}^n \lambda_j < \sum_{j=1}^m \mu_j$$

This constraint means that the average rate of jobs that all schedulers issue should never be faster than the average rate of jobs that all computing nodes execute.

$$\sum_{i=1}^n \lambda_j < a_{ij} \mu_j$$

This constraint means that average job rate sent to node j must not exceed rate at which jobs execute on j

μ_j is the average processing rate of jobs at the computational node j . λ_j is the average rate of jobs that scheduler i issues.

4.1. Objective Function to Optimize Task Response Time

Let $a_i = \{a_{i1}, a_{i2}, \dots, a_{im}\}$ represents scheduler i 's task slicing scheme on all cloud computing nodes where $i = 1, 2, \dots, n$ and a_{ij} is the task's ratio of jobs according to how scheduler i assigns jobs to node j . assuming that the average length of all tasks is b bits, transmission delay time from scheduler i to computing node j is e_{ij} , communication bandwidth from i to node j is c_{ij} , transmission time of task slice a_{ij} is calculated as follows:

$$L_{ij} = e_{ij} + \frac{b \cdot a_{ij}}{c_{ij}} \quad (8)$$

Average service time $E(t)$ is defined as:

$$E(t) = \frac{1}{\mu} + \frac{\lambda(\sigma^2 + \frac{1}{\mu^2})}{2(1 - \frac{\lambda}{\mu})} \quad (9)$$

Where σ^2 represents variance of the service time, μ is average execution rate of the computing node.

Assuming that the computing nodes service time subjects to the negative exponential distribution, the average service time F_{ij} of task slice a_{ij} can be computed as follows:

$$F_{ij} = \left(\frac{1}{\mu_j} + \frac{\sum_{k=1}^n \lambda_k a_{kj}}{\mu_j (\mu_j - \sum_{k=1}^n \lambda_k a_{kj})} \right) * a_{ij} \quad (10)$$

The execution time of task slice a_{ij} is the sum of transmission time L_{ij} and the average service time F_{ij} on node j as follows

$$F_{ij} + L_{ij} = \left(\frac{1}{\mu_j} + \frac{\sum_{k=1}^n \lambda_k a_{kj}}{\mu_j (\mu_j - \sum_{k=1}^n \lambda_k a_{kj})} \right) * a_{ij} + e_{ij} + \frac{b \cdot a_{ij}}{c_{ij}} \quad (11)$$

The slices are then executed independently after being scheduled to computing nodes. The response time of the task is:

$$FL_i(a_i) = \max_{j=1}^m (F_{ij} + L_{ij})$$

$$= \max_{j=1}^m \left(\left(\frac{1}{\mu_j} + \frac{\sum_{k=1}^n \lambda_k a_{kj}}{\mu_j (\mu_j - \sum_{k=1}^n \lambda_k a_{kj})} \right) * a_{ij} + e_{ij} + \frac{b \cdot a_{ij}}{c_{ij}} \right) \quad (12)$$

The scheduler expects minimum response time when scheduling tasks. Therefore, we can get the objective function of scheduler i (D_i) based on response time optimization:

$$D_i = \min \max_{j=1}^m \left(\left(\frac{1}{\mu_j} + \frac{\sum_{k=1}^n \lambda_k a_{kj}}{\mu_j (\mu_j - \sum_{k=1}^n \lambda_k a_{kj})} \right) * a_{ij} + e_{ij} + \frac{b \cdot a_{ij}}{c_{ij}} \right) \quad (13)$$

Introducing new variable μ_{ji} (*Computational power*) to enhance the equation's understandability:

$$\mu_{ji} = \mu_j - \sum_{k=1, k \neq i}^n \lambda_k a_{kj} \quad (14)$$

Combining 13 & 14 you get:

$$D_i = \min \max_j^m \left(\left(1 + \frac{\mu_j - \mu_{ji} + \lambda_{ji} a_{ij}}{\mu_{ji} - \lambda_{ji} a_{ij}} \right) * \frac{a_{ij}}{\mu_j} + e_{ij} + \frac{b * a_{ij}}{c_{ij}} \right) \quad (15)$$

5. Analysis

This computation schema proves to be better when coupled with the cloudlets computation approach in cloud-based mobile augmentation in mobile cloud computing technology. The computation schema also proves to be better compared with other algorithms used in solving the task allocation and response time challenges facing mobile cloud computing currently. Greedy algorithm utilized in cyber foraging approach compared with the proposed computation schema shows the effectiveness of cloudlet approach as opposed to the surrogate approach. Game-theoretic algorithm utilizes game theory to design new task scheduling algorithm with a goal of completion time to enhance the response time in the cloud.

The task proportion sent into a node is given by:

$$a_{ij} = \frac{\mu_{ji}}{\sum_{j=1}^m \mu_{ji}} \quad (16)$$

The relative arrival rate of jobs at scheduler i is represented by \emptyset_i with the average arrival rate calculated as follows

$$R_i = \emptyset_i * P \cdot \sum_{j=1}^m \mu_j \quad (17)$$

Where P is the required overall average systems load.

Taking the system load P to 0.5 average task length b to 1Mbits, average delay time between scheduler i to node j to 100Kbps, transmission delay time between scheduler i to node j (e_{ij}) to 0.5s.

This average task processing rate for computation schema will be labelled as A.

The average task processing rate Greedy algorithm for cyber foraging will be labelled B.

The average task processing rate Game theoretic algorithm will be labelled as C.

Table 1. Average Task Processing Rate of Computing Nodes

Computing node	1	2	3	4	5	6	7	8
A	0.28	0.22	0.19	0.23	0.20	0.26	0.22	0.23
B	0.26	0.19	0.18	0.20	0.18	0.23	0.22	0.20
C	0.22	0.15	0.15	0.17	0.12	0.19	0.18	0.17

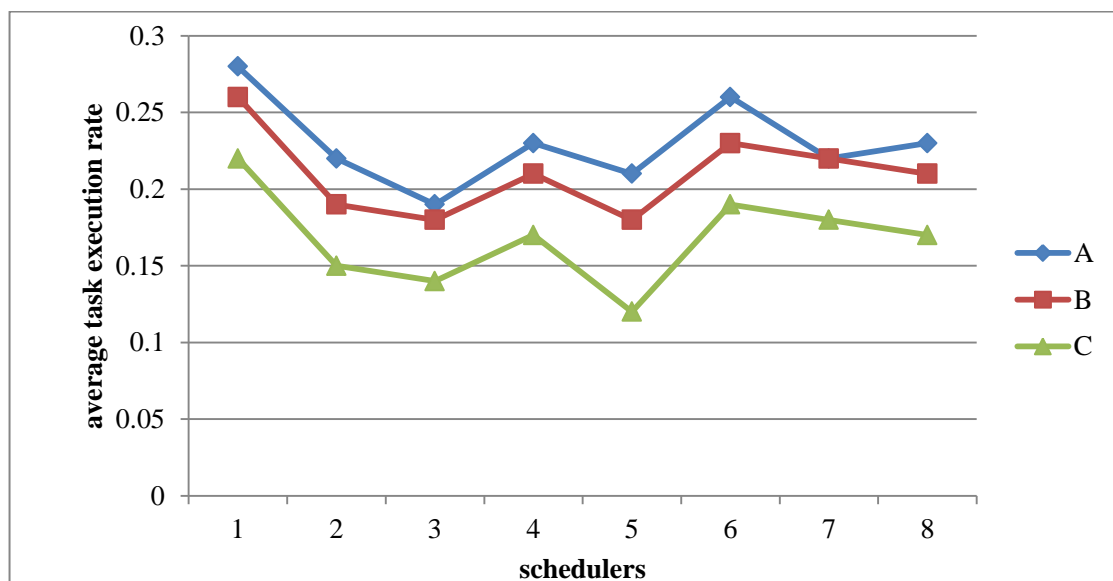


Figure 3. Analysis of the Average Task Processing Rates for A, B and C

6. Cloudlets Experimentation

An experiment was conducted by applying cloudlet model in cloud computing as in the figure 1. The cloudlets are executed on the local host to avoid overhead associated with cloudlet migration. An experimentation comparing two different scenarios, one involving with an equal number of virtual machines and cloudlets and another with VMs reduced to half the number of cloudlets.

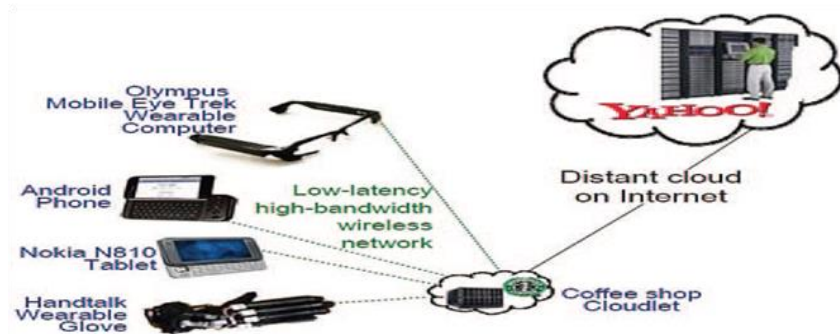


Figure 4: Concept of Cloudlets

The cloudlets are assigned to VMs by cloud service data centre brokers on the basis of the allocation policy. The average time consumption for the two scenarios is observed.

7. Results

This section shows the graphical and tabular representation of experimental results.

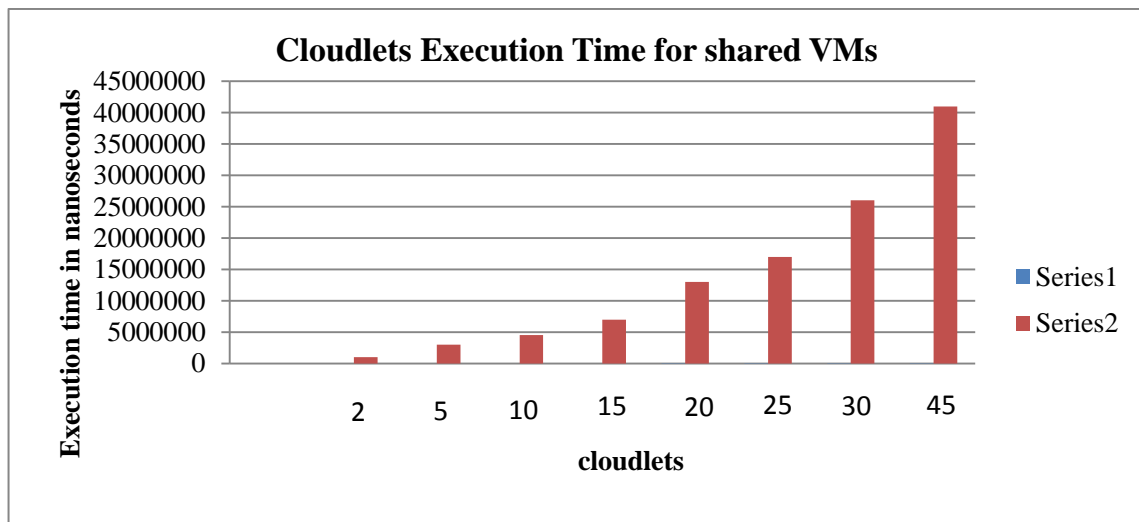


Figure 5. Cloudlets Execution Time for Shared VMs

The analysis shows that average execution time for each cloudlet increases with the equally increasing number of cloudlets and VMs. The execution time for cloudlet is about 58% for 2 -45 cloudlets.

Other experiments to ascertain percentage increase in average execution time for individual cloudlets in different experiments are as shown.

Table 2. Percentage Increase in the Average Execution Time of Cloudlet for Non-shared VMs

SNO	Cloudlets	% increase in Execution Time
	2-5	29
	5-10	55
	10-15	61
	15-20	56
	20-25	71
	25-30	66
	30-45	65

Table 3. Standard Deviation for Non-shared VM

SNO	Cloudlets	% Difference in STDEV
1	2-5	8
2	5-10	64
3	10-15	24
4	15-20	75
5	20-25	56
6	25-30	72
7	30-45	138

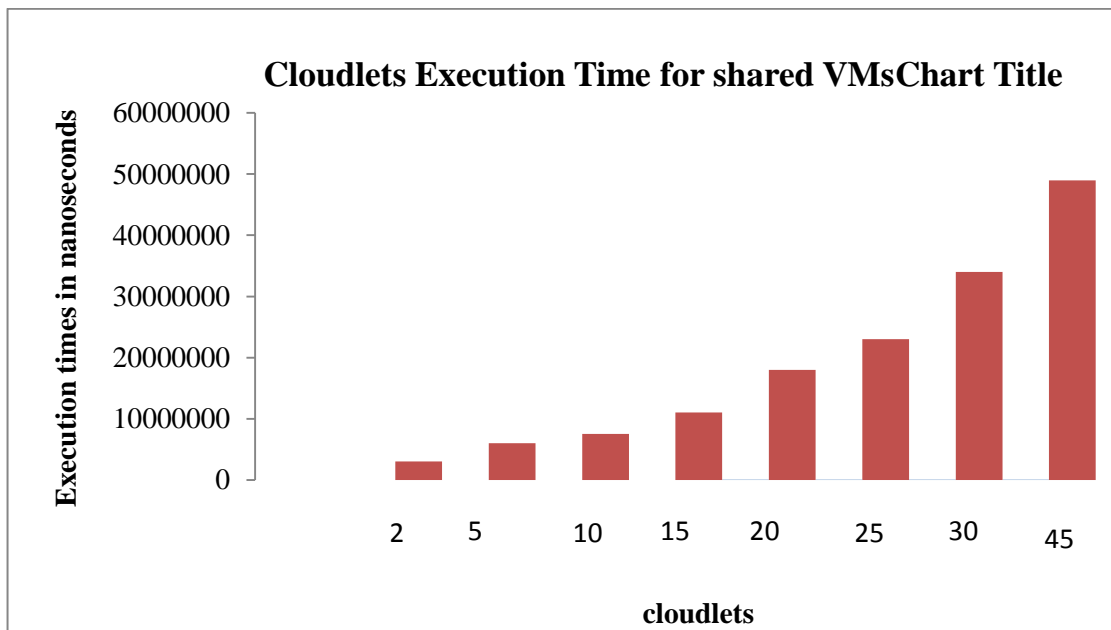


Figure 6. Cloudlets Execution Time for Shared VMs

The results indicate that average execution period for each cloudlet rises with the growth in the number of cloudlets regardless of the circumstance that the amount of VMs is reduced to half of the number of cloudlets. The average execution time for cloudlets increases by 64.7% for 2-45 cloudlets and 1-22 VMs

Table 4. Percentage Increase in Cloudlet Execution Time for Shared VM

SNO	Cloudlets	%Increase in Execution Time
1	2-5	32
2	5-10	81
3	10-15	60
4	15-20	64
5	20-25	77
6	25-30	69
7	30-45	70

Table 5. Percentage Difference in Standard Deviation of Cloudlet Execution Time for Shared VM

SNO	Cloudlets	% Difference in STDEV
1	2-5	79
2	5-10	118
3	10-15	32
4	15-20	91
5	20-25	64
6	25-30	74
7	30-45	135

8. Future Work

The Java RMI system in use is not adequately reliable. In case the server shuts down all the data and computation on the server gets lost. The client is able to resend the data and computation directions to another server. Improvement to this would boost great advancement on the technology.

Caching can be greatly applicable to cope with limited bandwidth and lost connectivity. Use of basic caching, live connection and piggy back fetching are common caching strategies. Use of different approaches for mobile client side caching can be used to enhance response time.

9. Conclusion

The paper shows a survey, classification and assessment of current available frameworks that enables development and utilization of cloud-based mobile augmentation in the mobile cloud computing technology. There are several differences in the existing and proposed frameworks for offloading resource-intensive workloads on the cloud infrastructure. Cloud computing seems to be a promising technology once fully executed eliminating vulnerabilities and delay time that exists. The paper analyses the cloud-based mobile augmentation on the cloud with a focus on delays and execution time. The execution time taken in the cloud to successfully work on a task requested or uploaded by a mobile device is of great importance in improving response time in the cloud. A task scheduling computation is proposed to enhance response time by breaking down the tasks into task slices. The utilization of cloudlets coupled with breaking down of large tasks into smaller tasks will greatly enhance the functionality of the cloud in terms of client's request and cloud's response times. Achievement of reduced delay time will greatly enhance the ability of cloud users to offload applications and information on the cloud whether partially or fully without fear of long delay time when those resources are required.

References

- [1] R. Khan, M. Othman, M. Ali, A. N. Khan and S. A. Madani, "Pirax: framework for application piracy control in mobile cloud environment", *Supercomput*, vol. 68, (2014), pp. 753–776.
- [2] M. Satyanarayanan, "Mobile computing: The next decade", *ACMSIGMOBILE Mob Comput CommunRev*, vol.15, no. 2, (2011), pp. 2–10.
- [3] S. Abolfazli, Z. Sanaei and A. Gani, "Mobile Cloud Computing", Research Lab, Faculty of Computer science and Information Technology University of Malaya MALAYSIA, (2012).
- [4] P. Dhivya and S. Uma, "Research In Computer Applications and Robotics", *Remote Display Solution For Mobile Computing*, (2014), Issn 2320-7345.
- [5] M. Ali, J. M. Zain, M. F. Zolkipli and G. Badshah, "Mobile Cloud Computing & Mobile Battery Augmentation Techniques: A Survey", (2009).
- [6] M. Satyanarayana, "Pervasive Computing: Vision and Challenges", *IEEE Personal Communications*, (2001), URL: <https://www.cs.cmu.edu/~aura/.../pcs01.pdf>.
- [7] N. Fernando, S. Loke and W. Rahayu, "Mobile cloud computing: A survey," *Future Generation Comput. Syst.*, vol. 29, no. 1, (2012), pp. 84–106.
- [8] S. Imai, T. Chestna and C. Varela, "Elastic Scalable Cloud Computing Using Application-Level Migration", *Proceedings of the 2012 IEEE/ACM 5th Int. Conf. on Utility and Cloud Computing*, (2012), pp. 91-98.
- [9] N. Tolia, D. G. Andersen and M. Satyanarayanan, "Quantifying interactive user experience on thin clients", *IEEE Comput.*, vol. 39, no. 3, (2006), pp. 46–52.
- [10] B. E. Nyamajejen and H. Yu, "Hurt of Bandwidth Cost on End Users Resources Consumption in Mobile Augmentation Cloud Based for Mobile Devices", vol. 3, iss. 4, (2014), www.ijsr.net.
- [11] R. Subrata, A. Y. Zomaya and B. A. Landreldt, "A Cooperative Game FrameWork for QoS Guided Job Allocation Schemes in Grids", *IEEE Transactions on Computers*, vol. 57, no. 10, (2008), 1413{14.
- [12] A. G. Delavar, M. Nejadkheirallah and M. Motalleb, "A New Scheduling Algorithm for Dynamic Taskand Fault Tolerant in Heterogeneous Grid Systems Using Genetic Algorithm", *Proceedings of the IEEE International conference on computer science and information technology*, vol. 9, (2010), 408-412.
- [13] R. Buyya, C. S. Yeo and S. Venugopal, "Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility", *Future Gener Comput Syst*, vol. 25, no. 6, (2009), pp. 599–616.
- [14] G. Orsinia, D. Badea and W. Lamersdor, "Context-Aware Computation Offloading for Mobile CloudComputing: Requirements Analysis, Survey and Design Guideline", *Procedia Computer Science*, vol. 56, (2015), pp. 10 – 17.
- [15] J. Shin, Y. Kim, W. Park and C. D. Park, "Cloud:a TPM- based secured at a access control method of cloud storage in mobile devices", *Proceedings of IEEE 4th international conference on cloud computing technology and science, (Cloud- Com12)*, Taipei, Taiwan, IEEE, (2012), pp. 551–600.
- [16] Z. Sanaei, S. Abolfazli, T. Khodadadi and F. Xia, "Hybrid Pervasive Mobile Cloud Computing: Toward Enhancing Invisibility", *Mobile Cloud Computing Lab, Faculty of CSIT, University of Malaya, Malaysia*, (2014).
- [17] P. Mell and T. Grance, "The NIST Definition of Cloud Computing (Draft)", U.S. Department of Commerce, National Institute of Standards and Technology, (2011).
- [18] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani and R. Buyya, "Cloud-Based Augmentation for Mobile Devices: Motivation, Taxonomies, and Open Challenges", *International Journal of Science and Research (IJSR)* ., ISSN (Online): 2319-7064, vol. 3, iss. 4, (2014).
- [19] H. Qi and A. Gani, "Research on Mobile Cloud Computing", *Review, Trend and Perspectives*, (2015).
- [20] T. Verbelen, T. Stevens, P. Simoens, F. Turck and B. Dhoedt, "Dynamic deployment and quality adaptation for mobile augmented reality applications", *J Syst Softw.*, vol. 84, no. 11, (2011), pp. 1871–82.
- [21] G. H. Canepa and D. Lee, "A virtual cloud computing provider for mobile devices", *Proceedings of the 1st ACM workshop on mobile cloud computing & services: social networks and beyond(MCS10)*, San-Francisco, CA, USA, ACM, (2010), pp. 1–5.
- [22] Z. Sanaei, G. A. Abolfazli and R. K. Buyya, "Heterogeneity in Mobile Cloud Computing: Taxonomy and Open Challenges", *IEEE Communications Surveys & Tutorials*, (2013).
- [23] M. Satyanarayanan, P. Bahl, R. Caceres and N. Davies, "The case for VM- based cloudlets in mobile computing", *Pervasive ComputIEEE*, vol. 8, no. 4, (2009), pp. 14–23.

Authors



ALshamaileh Mohamamd, he is from Jordan, his birth is in 12th October 1982 at City of Al Karak. Currently I'm a PhD student in College of Computer Science and Technology at Wuhan University of Technology. He received M.S in Software Engineering from South China University of Technology, Guangzhou China, in 2013. His research interests are mobile cloud computing.



Li Chunlin, she is a Professor of Computer Science in Wuhan University of Technology. She received the M.S in Computer Science from Wuhan Transportation University in 2000, and PhD in Computer Software and Theory from Huazhong University of Science and Technology in 2003. Her research interests include cloud computing and distributed computing.