Fault Tolerant Dynamic Task Offloading with Cloudlets in Mobile Cloud Computing

D. Shobha Rani, M. Pounambal

Abstract*—The mobile computing and cloud computing together form into Mobile Cloud Computing (MCC). The mobile devices have some limitations like computation power, battery life, etc., These limitations can be overcome by integrating with the cloud technology. Even in this case, delay will be introduced due to WAN properties during the offloading process of the tasks from mobile environment to cloud environment. This is reduced with the use of cloudlet environment. In this paper, fault tolerant dynamic task offloading with cloudlets in mobile cloud computing is proposed. The technical process of resisting against diseases by the human is incorporated in MCC to fight against the faulty machines in the system. The proposed system is simulated using openstack++ and evaluated in terms of task completion time. The performance of the proposed system is compared with the algorithm where rescheduling is not carried out and DR approach [28] and is proved to be performing better.

***Keywords:*—cloudlet, mobile cloud computing, disease resistance approach, fault tolerance.**

1. INTRODUCTION

Mobile Cloud Computing (MCC) is an integration of mobile computing and cloud computing technologies. Besides, mobile computing and cloud computing, it requires Internet also. Hence, it is the development of various technologies that are being grounded on Internet. It permits the users to enjoy the advantages of cloud computing technologies [1, 2]. In MCC, the cloud is used to provide resources for processing, storage, etc. In this process, the task to be processed need to be offloaded from mobile environment to cloud environment and vice-versa when task is completed [3, 4]. There are hurdles like energy consumption to enjoy these advantages to the fullest. These hurdles are because of the feeble communications in wireless technology [5]. The energy degrades during the waiting process for the hunt of wireless signals to communicate [6]. To conquer this issue, cloud computing and mobile devices are unified with each other to use the cloud infrastructure [27]. Generally, the cloud environment provides various services like Infrastructure as a Service (IaaS), Software as a Service (SaaS), Platform as a Service (PaaS), etc. In Mobile cloud computing, IaaS is being used.

The benefits of cloud computing, for example, high resource accessibility, ease and least energy utilization will support the mobile devices for offloading of the jobs and enhance the speed with which computation can be done. The job/task offloading is a key procedure in device environment where it needs to segment the given task into sub tasks and each assignment must be executed independently within or outside the device environment. On the off chance that the job is to be executed outside the device environment, i.e., the cloud, where resource scheduling plays a major role [28].

The notoriety and accessibility of mobile devices can enable versatile clients to enhance involvement of different administrations given by portable applications without the compel of time and place [17, 18]. Be that as it may, the resources of mobile devices are compelled. With constrained weight, size and battery life, the computational resources of mobile devices, for example, memory size and speed with which it can execute, are confined [19]. Since the calculation advancements on mobile devices are developing step by step, the execution of mobile devices still falls behind that of customary static PCs. Portable applications, for example, discourse recognizers [20, 21], regular dialect interpreters [22, 23], optical character recognizers [22] need more storage, processing speed, and lifetime of the device when compared to any other applications. Besides, contrasted and different applications that are at present reinforced on mobile devices, the applications specified above need quicker reactions.

The motivation for the introduction of mobile cloud computing is client server technology. As the client will have less number of resources like storage and processing capacity, similarly the mobile device has less storage capacity and less processing capability. The client uses the resources of the server to process the required tasks. It offloads the tasks to the server and the results are offloaded back to the client. Similarly, in mobile cloud computing, the mobile device offloads the task to the cloud environment where the tasks are processed and the results are offloaded back to the device environment. In this process, more number of hops might be required and the bandwidth may be low due to which the latency might be increased.

By utilizing framework, for example, Amazon's EC2 cloud, Microsoft Windows Azure or Rackspace, jobs that are computationally costly can be divested to the cloud. As this cloud environment is far from mobile devices, the underlying WAN leads to more latency which makes this methodology ineffective for ongoing applications. To address these problems, Satyanarayanan presented the idea of cloudlets [25], as reliable, PCs which are rich in resources and situated close to the versatile clients.

One of the answers for expanding the issues about mobile device restrictions can be cloud computing. In any case, WAN deferrals in the basic way of client cooperation can offended convenience by corrupting the freshness of framework reaction. Indeed, even unimportant client

Revised Manuscript Received on April 12, 2019.

D. Shobha Rani,School of Computing Science, and Engineering VIT, Vellore, Tamilnadu, India. (shobha24681@gmail.com)

Dr. M. Pounambal, School of Information Technology and Engineering VIT, Vellore, Tamilnadu, India. (mpounambal@vit.ac.in)

application collaboration acquire these deferrals in cloud computing. That is the motivation behind why cloudlet should be actualized. It is a trivial cloud positioned adjacent versatile clients which will be associated through LAN connect with clouds situated far away [26]. One of the advantages of cloudlet is certainly a probability that versatile clients can quickly initialize traditional virtual machines (VMs) on the cloudlet with the execution of needed application in a thin customer design.

A cloudlet is a portability upgraded little scale cloud datacenter that is situated nearer the Internet. The primary motivation behind the cloudlet is supporting resource-intensive and intelligent portable applications by giving intense processing resources to mobile devices with lower dormancy. It is another building component that expands the present cloud computing organization. It is amid mobile device environment and the Internet when a three-tier architecture is considered. The main objective of the cloudlet is make the advantages of the cloud computing closer to the mobile environment.

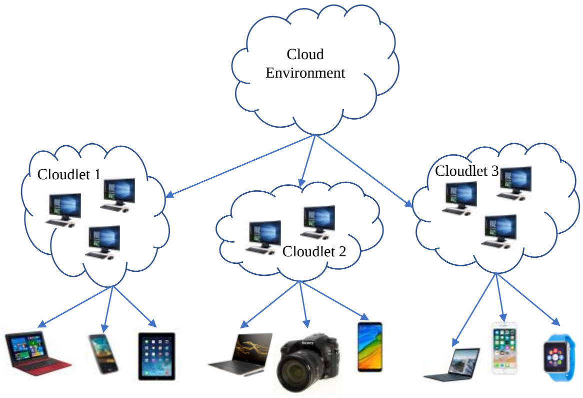

Fig. 1 Three Tier Architecture – Cloud, Cloudlet, Device Environment

The main variations of the cloudlet from the cloud environment are the speed with which the cloudlet is provided, handoff among cloudlets and detection of the cloudlet. The cloudlets are not static and hence the tasks need to be handoff from one cloudlet to another when the user associated with a particular cloudlet moves out of the corresponding range. At the same time, the mobile device need to detect, choose and link to the certain cloudlet among various existing cloudlets before provisioning [24].

The rest of the paper is organized as follows: section 2 presents the work done in this field, proposed work, fault tolerant dynamic task offloading algorithm with cloudlets in mobile cloud computing is discussed in section 3, section 4 projects the results and discussion is carried out and section 5 concluded the paper.

2. RELATED WORK

Numerous specialists and researchers have done different accomplishments in energy-aware MCC in past research. The exploration is assorted in alternate points of view [7]. Zhu and his group [8] grew continuous errands arranged virtualized distributed computing framework that was intended to accomplish vitality mindful planning for their ongoing works. The proposed arrangement [8] means to coordinate different vitality mindful planning calculations by utilizing a moving skyline improvement approach. In any case, this methodology did not think about versatility utilization and the comparable research concentrating on vitality mindful distributed computing frameworks has been expert by different researchers [9, 10, 11]. Besides, as one of the center procedures in distributed computing, virtual machine (VM) is viewed as a proficient methodology for working up cloud-based datacentre to accomplish green processing [12]. By and by, VM is just an administration portrayal approach that does not bring much specialized advancements, despite the fact that VM has been extensively connected in sending green Information Technology (IT) industry, for example, green information preparing, stockpiling, and transmissions [13, 14, 15].

In [16], a dynamic energy-aware cloudlet-based mobile cloud computing model (DECM) is proposed. It is concentrating on explaining the extra vitality utilizations amid the remote interchanges by utilizing dynamic cloudlets (DCL)-based model. A real situation is simulated and strong outcomes to the assessments are given in [16]. The fundamental commitments in [16] are two-overlay. The primary investigation of these authors is taking care of energy dissipate issues in the environment which is not static.

There are various planning ways to deal with take care of the generally standard issues in distributed computing [29-31]. The calculations have the ability to take care of the substantial issues and changing over them into sensible enhancement issues through direct programming [32], whole number straight programming [34], and number programming [33]. Because of the expansion in usefulness of the cloud administrations causes VM disappointments at the season of administration conveyance [45]. This sort of disappointment causes execution debasement, errand disappointments, information misfortune and misfortune in the income of the association [35]. While managing the work process of the errands, the undertaking disappointments causes basic issues. This is because of the execution of the work process relies upon the antecedent assignment, so the contribution of the undertaking was relying upon the yield of some other errand. In the ongoing years, numerous scientists focused on this territory [36, 37], however not very many are focused on the adaptation to non-critical failure system in the MCC condition.

There are two noteworthy adaptation to internal failure methodologies are famous under the cloud administrations are asset rescheduling and information replication. The information replication process is for the most part best at the season of booking of VMs and the asset rescheduling process is actualized at the season of assignment execution [38]. In the previous years, a few methods have been proposed for asset rescheduling [39-42]. Plankersteiner et al. [39] proposed a strategy for asset rescheduling by tending to the adaptation to internal failure. They presented the due date imperative for the undertaking took an interest in the planning procedure. Cao et al. [40] proposed three systems for asset rescheduling. On the off chance that the VM is smashed at the season of errand execution, this model consequently stores the executing assignment and sits tight for t seconds to repair of the VM. In the event that the VM isn't repaired, the errand moves to the next VM, generally the undertaking will execute on the equivalent VM. Stop et al. [49] proposed aggregate based instrument for adaptation to internal failure to accomplish dependable asset administration in MCC. This strategy considers the accessibility of asset in the cell phones and gatherings them as per the asset accessibility. The assets of cell phones are not steady, so the dynamic idea of the earth is considered for gathering of cell phones. They connected the adaptation to internal failure procedures utilizing the replication or check focuses to the gathering. Choi et al. [46] proposed adaptation to non-critical failure planning approach called as Content addressable system in MCC. This strategy coordinates the social computing-based condition for MCC.

The adaptation to non-critical failure planning is ordered with four sub modules, for example, cloud benefit conveyance, replication and load adjusting, QoS provisioning and malevolent client separating. This technique accomplished great outcomes for administration execution time and unwavering quality. Wadhwa et al. [47] talked about different sorts of deficiencies and adaptation to internal failure components in the cloud. The issues of accessibility and unwavering quality are tackled by utilizing the adaptation to non-critical failure instrument. They utilized the Nagios checking instrument for recognizing and breaking down different shortcomings. Cheraghlou et al. [48] endeavored a review on examination of adaptation to non-critical failure structures in distributed computing. The creators expressed the strategies of the adaptation to non-critical failure structures lastly they contrast the adaptation to non-critical failure methodologies and blame recognition limit and blame recuperation. Moreover, numerous calculations [41, 43, 44] are proposed for adaptation to internal failure in the disseminated condition regardless of rescheduling and replication. Barely any academicians focused on the most proficient method to foresee and deal with the deficiencies in the foundation which is useful to recognize the reasons for disappointments. In this paper, the engineering for adaptation to internal failure utilizing DR-approach was created. The progression as of late identified with the bio propelled approaches drawn the consideration of the world.

3. FAULT TOLERANT DYNAMIC TASK OFFLOADING WITH CLOUDLETS IN MOBILE CLOUD COMPUTING (FTDOCL)

The systems which are used for computation purpose either in the cloudlet environment or cloud environment are referred as virtual machines. When ever the task cannot be completed in the mobile environment, it is being offloaded to the cloudlet or cloud environment. At any point of time, the computational resources may fail. If tasks that require more computational resources than the VM has, or if the VM is assigned with more tasks than that can be computed, then VM might fail. These are referred as insufficient computational resources. If a particular machine does not support virtual machine functionalities, then it is referred as incompatible resources. Some tasks can be computed in some virtual machines only. i.e., the VM need to be compatible for executing the task. If not, an error is generated but the process will not be stopped and continue executing in the background. Hence, the corresponding VM will be not be able to accept any other new tasks also. This is referred as incompatible VM tasks. Hence, an effective fault tolerant technique is required to deal with the failures of the virtual machines.

The human body has resistance power to fight against bacteria, viruses, etc. In human body, the immune system has B-cells, T-cells and small white blood cells which help in attacking bacteria, toxins, cancerous cells, etc., hence, stop attack from various diseases. The human body has monitoring module which helps in enhancing the resistance power.

The mobile devices have limited resources like energy, computation power, etc. because of which the mobile device cannot perform large computations. Hence, offloading process is introduced. When a task T, is to be performed by a mobile device and if its own resources are not sufficient, then the task T_i is divided into 'n' subtasks T_{i1}, T_{i2}, T_{i3}, …, T_{in}, which enables parallel execution of the subtasks when submitted various VMs in the cloudlet or cloud environment. There will some amount of data that need to be communicated among the subtasks of a particular task. Three parameters considered corresponding to each virtual machine are the computation power, storage capacity, the execution speed. In cloudlet environment, each VM maintains monitoring module, memory module. In the cloud environment, both these modules are maintained in the server. In addition to these two modules, response module is also maintained.

Monitoring Module

As in the human body, in the case of cloudlet environment, monitoring module is maintained in every virtual machine which helps in identifying whether the VM is faulty or not. In the case of cloud environment, the monitoring module is maintained in the server which has many virtual machines and this monitoring module monitors the behaviour of each VM in the corresponding server. In case of cloudlet environment, the monitoring module sends the information to the mobile environment about the faulty VM. Then the corresponding mobile device offloads the task assigned to the faulty VM to another VM in the cloudlet environment. The monitoring module is also responsible to inform the mobile environment about the condition of the VM once it is repaired. If the task or subtask is not completed in estimated time, then the status message is requested from the corresponding VM and time limit is set by the monitoring module. If the VM responses with an acknowledgment within time limit, then the monitoring module will wait for some more time for the task or subtask to be completed. At this point, if the VM do not complete the task within extended time also, then it is considered to be faulty VM and is informed to the mobile device. If the

VM do not respond to the request of the status message then the information about the faulty message is sent to the mobile device. When the mobile device receives the information about the faulty VM, then the task is reassigned to another VM. The role played by the mobile device in the cloudlet environment is played by the server in case of the cloud environment. The computation power each VM is computed before assigning the task to it. If the monitoring module of the VM do not respond with the computation power when requested, then it is considered that the corresponding VM is faulty or crashed.

Response Module

In the case of cloud environment, the response module is responsible to reschedule the task from one VM to another of the same server and intimate the server about the faulty VM. The memory module of the server is responsible for storing the fixing technique of the problem. It means that the alternate VM which is similar to the faulty VM is used to fix the problem. Similarity is determined by the characteristics of the VMs. Hence, when VM_{ij} is determined as faulty, then which VM_{ik} is used to fix the problem is stored in the memory module, where i denotes the server, j and k denotes the VM associated with the corresponding server. This VM_{ik} is considered as the antibody of VM_{ij}. If more VMs are similar to VM_{ij}, then, preferable function determines which is better to offload the task from VM_{ij} to VM_{ik}.

The Preferable Function, $PF_{ij\rightarrow ik} = N \times \left(\frac{n}{BW_{ij\rightarrow ik}} + \frac{1}{CP_{ik}}\right)$, where N is the number of instructions to be offloaded from VM_{ij} to VM_{ik}, n is the number of bytes in instruction, CP_{ik} is the computation power of VM_{ik}.
Hence, the VM is chosen whose preferable function value is less.

If the information is not existing in the memory, then the response module transmits a message to all the VMs associated in the corresponding server and then offloading takes place.

The offloading can take place from the VM of one server to VM of another server also. This is via the mobile environment.

4. RESULTS AND DISCUSSION

Openstack++ is used to simulate the cloud and cloudlet environment and to simulate mobile or device environment, android mobile phones are used. Various scheduling approaches for different environments can be designed using Openstack++ because of its flexibility in deployment. The simulation is executed for 25 runs and the results are considered and graphs are plotted based on the average of these values. The simulator parameters used for the cloud setup is shown in Table 1.

Parameter	Value
Number of virtual machines	50-250
Number of instructions that a VM can execute	500-2500 MIPS
Memory Size	512-2048MB
Bandwidth	0.5-200 Mbps
Number of Tasks	500

Initially, the cloud needs to be setup and the task settings need to be initialized. The number of servers is setup and then the network among these servers is established. The network properties are defined in the simulator. Then, number of tasks, task size, number of virtual machines for each server and the computation power of each virtual machine are defined.

The fault rate considered to be varying for the simulation of the proposed algorithm, FTDOCL. The parameter considered for the evaluation of the performance of the algorithm is the time in which the assigned tasks are completed.

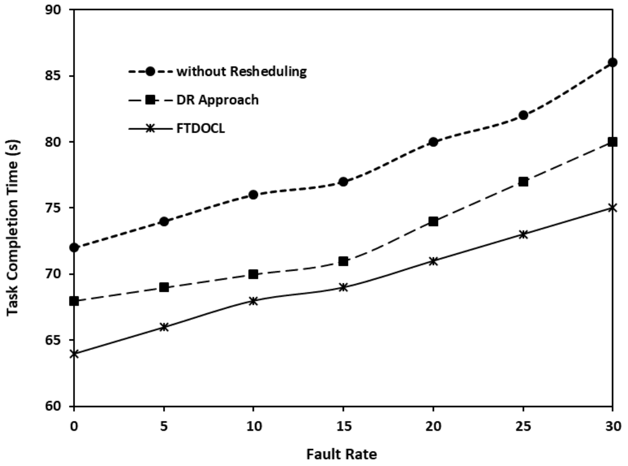

Fig. 2 Fault Rate vs Task Completion Time

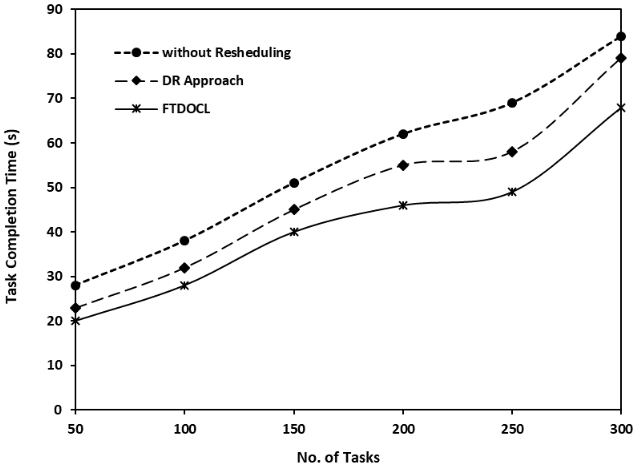

Fig. 3 Number of Tasks vs Task Completion Time

Fig. 2 shows the performance comparison for the algorithm without rescheduling, DR approach and the proposed algorithm, FTDOCL in terms of task completion time with respect to fault rate. It can be observed that the proposed algorithm performs better when compared to the legacy systems. Also, the increase in the task completion time is slow as there is increase in fault rate in the case of the proposed algorithm, FTDOCL. The number of tasks considered in this case is 250. The size of each task considered is approximately 150 KB. The performance of the proposed algorithm is enhanced by 12.5% when compared with the algorithm without rescheduling and 5% when compared with the DR approach.

Fig. 3 shows the performance of the proposed algorithm, FTDOCL in terms of number of tasks with respect to task completion time. It is compared with the algorithm where rescheduling is not carried out and the disease resistance approach. The proposed algorithm, FTDOCL performs better as shown in the Fig. 3. The performance is enhanced because of the cloudlet environment. The performance of the algorithm is enhanced by 32.3% when compared with the algorithm without rescheduling and 16.33% when compared with the DR approach.

Fig. 4 shows the performance of the proposed algorithm, FTDOCL in terms of number of tasks with respect to task completion time for varying fault rates. The proposed algorithm, FTDOCL performs better when fault rate is less as shown in the Fig. 3. The performance of the algorithm is reduced only by 3.3% when the fault rate is increased from 10% to 15% and 6.5% when the fault rate is increased to 20%.

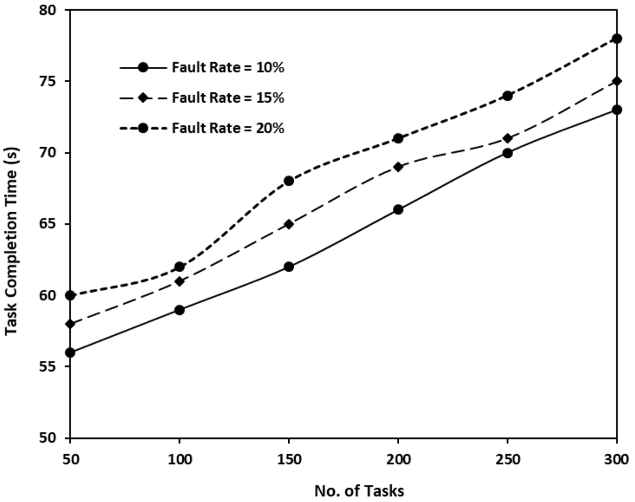

Fig. 4 Number of Tasks vs Task Completion Time

5. CONCLUSION

In this paper, an algorithm is proposed to handle the faulty virtual machines in the MCC with cloudlets. The mechanism of resisting towards the diseases by human is the basis for this algorithm. The process is considered separately when the virtual machines are faulty in the case of cloudlet environment and cloud environment. In case of cloud environment, three modules namely monitoring module, memory module and response module are designed. In case of cloudlet environment, only two modules namely monitoring and memory modules are maintained. The task being performed by the response module in case of cloud environment is performed by the device in case of cloudlet environment. The performance is measured in terms of task completion time by simulating the proposed algorithm using openstack++ simulator. The performance of the proposed algorithm is compared with legacy system like DR approach and proved to be performing better.

REFERENCES

1. Sabharwal M, Agrawal A, Metri G, Enabling green IT through energy-aware software, IT Prof 2013:19–27. doi: http://doi.ieeecomputersociety.org/10.1109/MITP.2012.104.
2. Bonino D, Russis L D, Corno F, Ferrero G, JEERP: energy aware enterprise resource planning. ITProf2013;50–6. doi: http://dx.doi.org/10.1109/MITP.2013.22.
3. Huang D, et al. Mobile cloud computing. IEEE COMSOC Multi med Commun Tech Comm (MMTC)E-Lett2011; 6(10): 27–31.
4. Kumar K, Lu Y. Cloud computing for mobile users: can offloading computation save energy?. Computer2010;43(4):51–6. http://dx.doi.org/10.1109/MC.2010.98.
5. Guan L, Ke X, Song M, Song J. A survey of research on mobile cloud computing. In: Proceedings of the 10th IEEE/ACIS international conference on computer and information science. IEEE Computer Society; 2011.p.387–92. doi: http://doi. ieeecomputersociety.org/10.1109/ICIS.2011.67.
6. Han B, Hui P, Srinivasan A. Mobile data offloading in metro politan area networks. ACM SIG MOBILE Mob ComputCommun Rev2011; 14(4):28–30. http://dx.doi. org/10.1145/1942268.1942279.
7. Gupta S, Roy K. Comparison of different energy minimization techniques in wireless sensor network. Energy 2013;75(18), http://dx.doi.org/10.5120/13348-0576.
8. Yang L, Zhu X, Chen H, Wang J, Yin S, Liu X, “Real-time tasks-oriented energy-aware scheduling in virtualized clouds”, IEEE Trans Cloud Comput 2014; 2(2): 168–80. http://dx.doi.org/10.1109/TCC.2014.2310452.
9. Mezmaz M, Melab N, Kessaci Y, Lee Y C, Talbi E, etal. “A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems. J Parallel DistribComput 2011;71(11): 1497–508. http://dx.doi.org/10.1016/j. jpdc.2011.04.007.
10. Beloglazov A, Buyya R. Energy efficient resource management in virtualized cloud data centers. In: Proceedings of the 10th IEEE/ACM international conference on cluster, cloud and grid computing. IEEE Computer Society; 2010. pp. 826–31.
11. Berl A, Gelenbe E, Girolamo M D, Giuliani G, Meer H D, Dang M Q, etal. Energy-efficient cloud computing. Comput J 2010; 53(7): 1045–51. http://dx.doi.org/ 10.1093/comjnl/bxp080.
12. Aksanli B, Venkatesh J, Rosing T. Using data center simulation to evaluate green energy integration. Computer 2012; 45(9) 0056–64, doi: http://doi.ieeecompu tersociety.org/10.1109/MC.2012.249.
13. Baliga J, Ayre R W, Hinton K, Tucker R. Green cloud computing: balancing energy in processing, storage, and transport. Proc IEEE2011; 99(1):149–67. http://dx.doi. org/10.1109/JPROC.2010.2060451.
14. Lee Y C, Zomaya A Y. Energy efficient utilization of resources in cloud computing systems. J Super comput 2012; 60(2): 268–80. http://dx.doi.org/10.1007/s11227- 010-0421-3.
15. Xiao Z, Song W, ChenQ, “Dynamic resource allocation using virtual machines for cloud computing environment”, IEEE Trans Parallel DistribSyst 2013; 24 (6): 1107–17. http://dx.doi.org/10.1109/TPDS.2012.283.
16. Gai, K., Qiu, M., Zhao, H., Tao, L., &Zong, Z. (2016). Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing. *Journal of Network and Computer Applications*, *59*, 46-54.

17. Sanaei, Z., Abolfazli, S., Gani, A., Buyya, R., 2014. Heterogeneity in mobile cloud computing: taxonomy and open challenges. IEEE Communications Surveys & Tutorials 16 (1), 369–392
18. Fu, S., Atiquzzaman, M., Ma, L., Lee, Y.-J., 2005a. Signaling cost and performance of sigma: A seamlesshandover scheme for data networks. Wireless Communications and Mobile Computing 5 (7), 825–845.
19. Shaukat, U., Ahmed, E., Anwar, Z., Xia, F., 2016. Cloudlet deployment in local wireless networks: Motivation, architectures, applications, and open challenges. Journal of Network and Computer Applications 62, 18–40.
20. Kristensen, M. D., 2007. Enabling cyber foraging for mobile devices. In: Proceedings of the 5th MiNEMAWorkshop: Middleware for Network Eccentric and Mobile Applications. pp. 32–36.
21. Su, Y.-Y., Flinn, J., 2005. Slingshot: deploying stateful services in wireless hotspots. In: Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services. ACM, pp. 79–92.
22. Balan, R. K., Gergle, D., Satyanarayanan, M., Herbsleb, J., 2007. Simplifying cyber foraging for mobiledevices. In: Proceedings of the 5th International Conference on Mobile Systems, Applications and Services.ACM, pp. 272285.
23. Flinn, J., Park, S., Satyanarayanan, M., 2002. Balancing performance, energy, and quality in pervasivecomputing. In: Proceedings of the 22nd International Conference on Distributed Computing Systems. IEEE, pp.217–226.
24. Ha, K., &Satyanarayanan, M. (2015). Openstack++ for cloudlet deployment. School of Computer Science Carnegie Mellon University Pittsburgh.
25. M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The case for vm-based cloudlets in mobile computing,” Pervasive Computing, IEEE, vol. 8, no. 4, pp. 14–23, 2009
26. [3] E. Koukoumidis, D. Lymberopoulos, K. Strauss, J. Liu, and D. Burger, “Pocket cloudlets,” ACM SIGPLAN Notices, vol. 47, no. 4, pp. 171–184, 2012
27. Rahimi MR, Ren J, Liu CH, Vasilakos AV, Venkatasubramanian N. “Mobile cloud computing: A survey, state of art and future directions”, Mobile Networks and Applications. pp. 133-143, 2014.
28. Dasari, N. R., &Saritha, V. (2016). Architecture for fault tolerance in mobile cloud computing using disease resistance approach. International Journal of Communication Networks and Information Security (IJCNIS), 8(2).
29. El Zant, Bassem, Isabel Amigo, and Maurice Gagnaire. “Federation and revenue sharing in cloud computingenvironment.” In Proceedings of the IEEE International Conference on Cloud Engineering. pp. 446–451, 2014
30. Chen Shi, J. Wu, and Z. H. Lu. “A cloud computing resource scheduling policy based on genetic algorithm with multiple fitness.” In Proceedings of the IEEE 12th International Conference on Computer and Information Technology. pp. 177–184, 2012.
31. Chen, N., Chen, W.N., Gong, Y.J., Zhan, Z.H., Zhang, J., Li, Y. and Tan, Y.S. "An evolutionary algorithm with doublelevel archives for multiobjective optimization.” IEEE transactions on cybernetics, vol. 45, No.9, pp.1851-1863, 2015.
32. Kumar, Senthil SK, and P. Balasubramanie. “Dynamic scheduling for cloud reliability using transportation problem.” Journal of Computer Science vol. 8, No. 10 pp. 1615–1626, 2012.
33. Li, Qiang, and YikeGuo. “Optimization of resource scheduling in cloud computing.” In Proceedings of the 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing. pp. 315–320, 2010.
34. Genez, Thiago AL, Luiz F. Bittencourt, and Edmundo RM Madeira. “Workflow scheduling for SaaS/PaaS cloud providers considering two SLA levels.” In Proceedings of the IEEE Network Operations and Management Symposium, pp. 906–912, 2012.
35. Smanchat, Sucha, and KanchanaViriyapant. “Taxonomies of workflow scheduling problem and techniques in the cloud.” Future Gener. Comput. Syst. Vol. 52, pp.1–12, 2015.
36. Lee, Young Choon, Hyuck Han, Albert Y. Zomaya, and MazinYousif. “Resource-efficient workflow scheduling in clouds.” Knowl.-Based Syst. Vol. 80, pp. 153–162, 2015.
37. Ramezani F., Lu J., Taheri J. “Evolutionary algorithm-based multi-objective task scheduling optimization model in cloud environments.” World Wide Web: pp.1–21, 2015.
38. Zhu, Xiaomin, Chuan He, RongGe, and Peizhong Lu. “Boosting adaptivity of fault-tolerant scheduling for real-time tasks with service requirements on clusters.” J. Syst. Softw, pp.1708–1716, 2011.
39. Plankensteiner, Kassian, and RaduProdan. “Meeting soft deadlines in scientific workflows using resubmission impact”. IEEE Trans. Parallel Distrib. Syst. pp. 890–901, 2011.
40. Cao, Yang, CheulWoo Ro, and JianWei Yin. “Scheduling Analysis of failure-aware VM in cloud system.” Int. J. Control Autom. Vol. 7, No.1, pp. 243–250, 2014.
41. Chen, Wei, Young Choon Lee, Alan Fekete, and Albert Y. Zomaya. “Adaptive multiple-workflow scheduling with task rearrangement.” J. Supercomput. pp. 1–21, 2015.
42. Olteanu, Alexandra, Florin Pop, CiprianDobre, and ValentinCristea. “A dynamic rescheduling algorithm for resource management in large scale dependable distributed systems.” Comput. Math. Appl. pp. 1409–1423, 2012.
43. Guan, Qiang, Ziming Zhang, and Song Fu. “Ensemble of bayesian predictors and decision trees for proactive failure management in cloud computing systems.” J. Commun. pp. 52–61, 2012.
44. Rood, Brent, and Michael J. Lewis. “Grid resource availability prediction-based scheduling and task replication.” J. Grid Comput. pp. 479–500, 2009.
45. Javadi, Bahman, Daishi Kondo, Jean-Marc Vincent, and David P. “Discovering statistical models of availability in large distributed systems: An empirical study of seti@ home.” IEEE Trans. Parallel Distrib. Syst. pp.1896–1903, 2009.
46. Choi, S., Chung, K., & Yu, H. “Fault tolerance and QoS scheduling using CAN in mobile social cloud computing.” Cluster Computing, *17*(3), pp. 911-926, 2014.
47. Wadhwa, A., &Bala, A. “Preventing Faults: Fault Monitoring and Proactive Fault Tolerance in Cloud Computing.” 118 International Journal of Communication Networks and Information Security (IJCNIS) Vol. 8, No. 2, August 2016 In Proceedings of International Conference on ICT for Sustainable Development, pp. 665-673, 2016
48. Cheraghlou, M. N., Khadem-Zadeh, A., &Haghparast, M. “A survey of fault tolerance architecture in cloudcomputing.” Journal of Network and Computer Applications, vol. *61,* pp. 81-92, 2016.
49. Park, J., Yu, H., Kim, H., & Lee, E. “Dynamic group-based fault tolerance technique for reliable resource management in mobile cloud computing.” Concurrency and Computation: Practice and Experience, 2014.