

Mobile Cloud Computing: Critical Analysis of Application Deployment in Virtual Machines

Muhammad Shiraz¹, Abdullah Gani².

¹ muh_shiraz@siswa.um.edu.my

Faculty of Computer Science and Information Technology University of Malaya,
Kuala Lumpur Malaysia

²abdullah@um.edu.my

Faculty of Computer Science and Information Technology University of Malaya,
Kuala Lumpur Malaysia

Abstract. The proliferation of smart mobile devices and incredible deployment of computing as utility in mobile cloud computing leads to the access of computational intensive applications on smart mobile devices. Diverse performance metrics are associated with the execution of applications (cloudlets) in distributed mobile cloud computing environment; such as cloudlet offloading, cloudlet allocation to VM, cloudlet scheduling in VM, cloudlet migration in datacenter and cloudlet reintegration. In this paper we critically investigate the key performance metrics associated with the deployment of VM in execution of cloudlets in cloud computing. We perform experimentation in simulation environment using CloudSim. We analyze the impact of performance metrics on the execution of cloudlets. Variant performance metrics associated with the deployment of VM affect the cloudlet execution; for that reason it is mandatory to efficiently manage the deployment of VMs in cloud infrastructure and exploit optimal techniques for distributed processing to minimize the overhead associated with cloudlet execution.

Keywords: Mobile Cloud Computing, Virtual Machines, Distributed Applications, Cloudlets, Lightweight

1 Introduction

The proliferation of smart mobile devices, the incredible deployment of enormous computing resources and the vision of computing as utility [1] brought a tremendous revolution in the information processing world. Cloud revolution leads to the access of computational intensive applications on smart mobile devices. In maintaining MCC, researchers use diverse approaches worldwide to augment computing potentials of smart mobile devices. The predominant approach is application offloading, in which smartphones offload computation intensive or data intensive applications to VMs running in datacentre. Diverse performance metrics are associated with the execution of applications (cloudlets) in distributed mobile cloud computing environment. In cloud computing paradigm; applications run on virtual machines (VMs) and the performance relies on efficient management of the deployment of VMs in their entire life cycle. A VM serves as an individual virtual computing element and all VMs in a host share the same CPU/core which increases the CPU scheduling latency for each VM significantly [2]. The miniature nature and resources limitations of smart mobile devices crave for lightweight and efficient distributed application framework with minimum possible resources consumption and maximum possible throughput on smart mobile devices.

Corresponding Author: Tel.: +60 (0102927937).
E-mail address: muh_shiraz@siswa.um.edu.my

In this paper we analyse the impact of diverse performance metrics in the deployment of VMs for the execution of cloudlets in cloud computing paradigm. We perform experimentation in CloudSim; a simulator for the design and evaluation of cloud computing infrastructure. We highlight the overhead involved in the execution of application in cloud computing infrastructure.

The paper is organized as; section 2, summarizes existing distributed application frameworks in mobile cloud computing; section 3, spotlights experimentation, results and discussion and section 4, draws conclusive remarks.

2 Distributed Application Execution Platforms

In maintaining mobile cloud computing, researchers use diverse approaches worldwide to augment computing potentials of smart mobile devices. The predominant approach is application offloading, in which elastic mobile application is partially or completely offloaded to remote resources rich cloud servers. Existing distributed application frameworks deploy distributed execution platform in local pervasive/ubiquitous environment or cloud server based centralized monitoring environment. Process offloading frameworks accomplish offload processing in diverse modes. Several approaches exploit VM cloning and entire application migration [3-8]; others focus on part(s) of the application to be offloaded. A number of approaches employ static partitioning [3,9]; others exercise dynamic partitioning. Variant migration patterns are used; VM transfer, downloading application by providing URL to remote host, Mobile agent such as USMC, application binary transfer and use of proxies. Diverse objective functions are considered; saving processing power, efficient bandwidth utilization, saving energy consumption, user preferences, and execution cost. Objective of all approaches is to augment the computing potentials of resources constraint mobile devices. The common aspect of the frameworks is that distributed execution platform is established at runtime which consumes computing resources of mobile devices copiously.

3 Experimentation, Results and Discussion

In cloud computing IaaS; applications run on virtual machines (VMs) which engrosses extra overhead in VM deployment. We conduct diverse experiments for the investigation of performance metrics associated with the execution of cloudlets in cloud computing infrastructure. We investigate the overhead associated with cloudlet allocation to VM and cloudlet execution. We spotlight the impact of VM deployment in the execution of cloudlets in diverse scenarios and analyze the overhead associated with the execution of cloudlets in VMs.

3.1 Cloudlet Execution

A cloudlet models the cloud based application services such as: content delivery, social networking, and business workflow. Cloudlet execution life cycle includes cloudlet creation, cloudlet allocation to VM, cloudlet scheduling in VM, and cloudlet termination. In some scenarios cloudlet migration also occurs in which cloudlets are migrated to other VMs at runtime by deploying variant migration policies. Cloudlet migration includes the overhead of saving state of running cloudlet, selection of appropriate remote host, transferring cloudlet to the remote host and allocation of cloudlet to a new VM in remote datacenter.

In our experimentation all the cloudlets are executed on the local host. For that reason, it does not involve the overhead associated with cloudlet migration. We conduct experiments in two different scenarios; first we maintain equal number of VMs and cloudlets, as a result a single cloudlet is executed on a single VM. In second, scenario we reduce the number of VMs to half of the number of cloudlets; in either scenario we conclude with different results.

3.1.1 Cloudlet Allocation to VM

In IaaS; cloudlets are allocated to VMs by datacenter broker on the basis of cloudlet allocation policy. We experiment the overhead associated with the allocation of a cloudlet to VM by creating equal number of cloudlets and virtual machines. We observe the average time consumed for the allocation of cloudlet to VM for equally increasing number of VMs and cloudlets. In all the experiments, the specification of cloudlets

and VMs are maintained homogenous. Fig. 1 illustrates increase in the average time required for the allocation of cloudlets to VMs in diverse experiments.

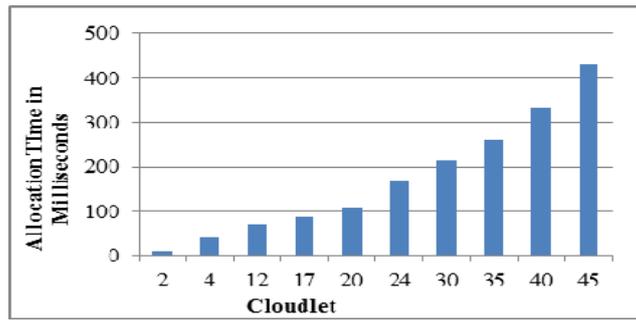


Fig. 1: Cloudlet Allocation to Virtual Machine Time Analysis

Analysis of the results indicates that the average time required for allocating of cloudlets to VMs increases with the increase in number of cloudlets and VMs, which indicates the overhead associated with cloudlet allocation to VM increases with the increase in number of cloudlets and VMs.

3.1.2 Cloudlets Execution Time for Equal Number of VMs and Cloudlets

The execution time of applications (cloudlets) changes with variant number of VMs. We conduct 8 experiments for equally increasing number of cloudlets and VMs. A single VM is allocated a single cloudlet and there is no overhead of VM scheduling for cloudlets. Each experiment is conducted five times to derive the precise average execution time of a single cloudlet. Fig. 2 indicates the average execution time of cloudlets.

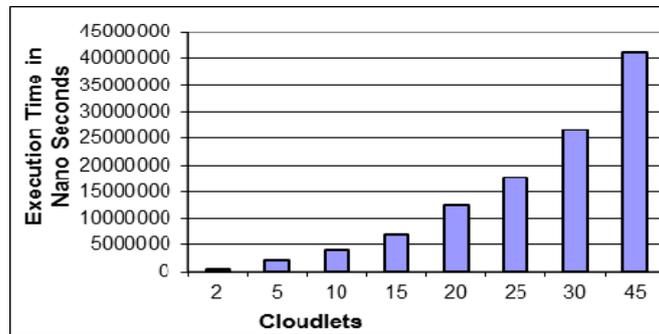


Fig. 2: Cloudlet Average Execution Time Analysis

Analysis of the results indicates that average execution time for each cloudlet increases with the equally increasing number of cloudlets and VMs; on average the execution time of cloudlet increases by 57% for 2-45 cloudlets. Table 1, shows the percentage increase in the average execution time for single cloudlet in different experiments.

Table 1: Percent Increase in the Average Execution Time of Cloudlet for Non-Shared VMs

SNO	Cloudlets	% Increase in Execution Time
1	2-5	28
2	5-10	54
3	10-15	60
4	15-20	55
5	20-25	71
6	25-30	66
7	30-45	65

Table 2 shows difference in standard deviation for cloudlet execution time in diverse experimentation.

Table 2 Percent Difference in Standard Deviation of Cloudlet Execution Time 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

3.1.3 Cloudlets Execution Time for Shared VMs

We conduct 8 experiments for increasing number of cloudlets, whereas maintaining the number of VMs half to the number of cloudlets, for this reason a single VM is allocated one or more cloudlet. Each experiment is conducted five times to derive the precise average execution time value for cloudlet. Fig. 3 indicates the average execution time of a single cloudlet for shared VM scenario.

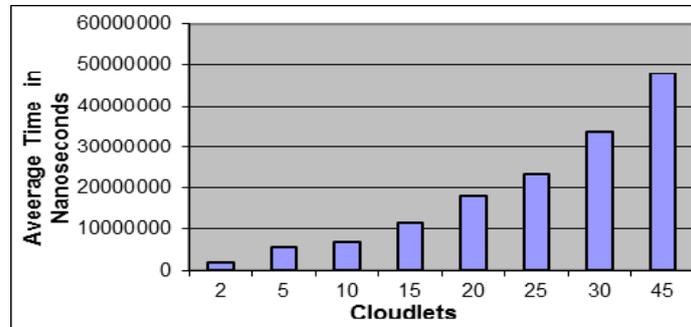


Fig. 3: Cloudlets Average Execution Time Analysis with Shared VMs

Analysis of the results indicates that average execution time for each cloudlet increases with the increase in the number of cloudlets despite the fact that the number of VMs is reduced to half of the number of cloudlets; on average the execution time of cloudlet increases by 64.7% for 2-45 cloudlets and 1-22 VMs. Table 3, shows the increase in the average execution time for single cloudlet in different experiments.

Table 3: Percent 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 4 shows difference in Standard deviation for cloudlet execution time in diverse experiments with shared VMs.

Table 4: Percent 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

3.1.4 Synthesis of Cloudlet Execution

In section 3.1.2, we highlight the impact of VMs on average cloudlet execution time by creating separate VM for each cloudlet. In section 3.1.3, we analyze the average execution time for cloudlet executed in shared VM. Fig.4 shows the synthesis of average cloudlets execution time in diverse scenarios.

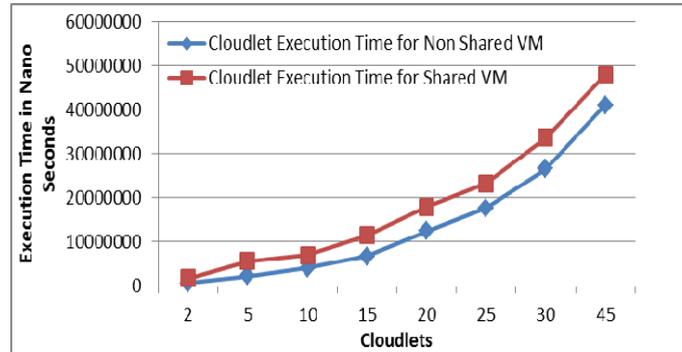


Fig. 4: Synthesis of Cloudlets Average Execution Time for Shared VMs and Non-Shared VMs

The comparison of cloudlet execution time in diverse scenarios concludes that average execution time of the cloudlet increases in either scenario. Further investigation in the increased value for either scenario establishes that average execution time of the cloudlet is higher with shared VMs as compare to non-shared VMs; for the reason that some extra overhead is associated with the scheduling of VMs among multiple cloudlets for shared VMs.

4 Conclusion

We analyzed diverse performance metrics associated with the deployment of VMs in the execution of cloudlets in cloud computing infrastructure. We conclude that variant performance metrics increase the cloudlet execution time; such as deployment of VM in the execution of cloudlet which involves the overhead of VM life cycle, cloudlet allocation to VM and scheduling of VM. For that reason it is mandatory to efficiently manage deployment of VMs in cloud infrastructure and exploit optimal techniques for distributed processing to minimize the overhead associated with cloudlet execution. The miniature nature and intrinsic limitations associated with smart mobile devices crave to deploy lightweight distributed application framework to easily deploy the distributed environment and efficiently establish distributed platform.

5 References

- [1] R. Buyya, S.C. Yeo, S. Venugopal, J. Broberg, I. Brandic “Cloud Computing and Emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility” *Future Generation Computer Systems*, Vol. 25, No: 6, pp. 599-616 2009.
- [2] Kangarlou, S. Gamage, R.R. Kompella, D. Xu “vSnoop: Improving TCP Throughput in Virtualized Improving TCP Throughput in Virtualized Environments via Acknowledgement Offload”, SC10 New Orleans, Louisiana, USA, IEEE Computer Society 2010.
- [3] B.G. Chun, P. Maniatis Augmented Smartphone Applications Through Clone Cloud Execution, Intel Research Berkeley 2009.
- [4] M. Satyanarayanan, P. Bahl, R. Caceres The Case for VM-Based Cloudlets in Mobile Computing IEEE Computing Society. October–December 2009.
- [5] B.G. Chun, S.Ihm, P.Maniatis, M.Naik, A. Patti “CloneCloud: Elastic Execution between Mobile Device and Cloud” *EuroSys’11 Salzburg Austria ACM Press*, April 10–13, 2011
- [6] B. Zao, Z. Xu, C. Chi, S. Zhu, G. Cao “Mirroring Smartphones for Good: A Feasibility Study” *ZTE Communications Volume:9: No:1 pp:13-18 March 2011*.

- [7] H. S. Hung, T.W. Kuo, C.S.Shih, J.P. Sheih, C.P. Lee, C.W. Chang, J.W. Wei A Cloud Based Virtualized Execution Environment for Mobile Applications ZTE Communications Volume: 9 No:1 pp: 19-25, March 2011.
- [8] S. Goyal, J. Carter A Lightweight Secure Cyber Foraging Infrastructure for Resource-Constrained Devices, WMCSA 2004 Sixth IEEE Workshop, IEEE Publisher, 2-3 Dec. 2004.
- [9] A. Dou, V. Kalogeraki, D. Gunopulos, T. Mielikainen, V. H. Tuulos “Misco: A MapReduce Framework for Mobile Systems”, PETRA’10 Samos, Greece. ACM Press, , June 23 - 25, 2010.