International Journal of Engineering & Technology, 7 (4.19) (2018) 177-181

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

QOS Aware Self Adaptable Virtual Machines Management System for Cloud Computing

J.Selvin Paul Peter[1], G. Mahadevan[2], S Selvakumar[3]

[1]*Research Student, AMCRC*
[2]*Professor, AMCEC, Bangalore*
[3]*Assistant Prof, SRMIST, Chennai*

Abstract

Cloud Computing as of now is the most important distributed environment because of low level user management and system integration. But most important challenge cloud computing faces is effective resource provisioning, Solving the issue will result in effective consumption of service offered, better user satisfaction and resources for more people during peak hours, reduce operational burden to cloud service providers and less pay to clients. Current works are aimed at determining the usage, VM (Virtual Machine) establishment and setting up. The above process requires considerable time to construct and kill VMs which may be used to cater more user. So here we have provided, a Quality of Service Aware Virtual Machine management mechanism for creating new VM's that makes use of the system resources efficiently. The existing VM for related type of requests are identified to minimize VM creation time. In our system, QOS is guaranteed by making all tasks adhere to the SLA necessities. Services are divided using need of the hour and the critical job is given higher significance. The experimental results show that a large number of users are serviced in relation to others algorithm which will fulfil clients needs during the peak traffic.

Keywords:

1. Introduction

Cloud Computing has developed as the efficient distributed environment which has attracted the interest of academic establishments, government and industries. Cloud computing service provides provide various types of services to clients such as software, storage and application over the Internet as means for creating customer friendly platform etc. Cloud computing services can be accessed by its users via different types of devices such as Personal Computers, mobile phones, notebooks, handhelds etc.

Cloud computing services are more trustworthy than services provided by grid computing. However, cloud computing still faces large amount of obstacles like an effective system for virtual machine creating which will guarantee QoS. Realizing QOS includes a wide range of constraints such as packet loss, bandwidth, Jitter, cost and reputation etc. and also factors like security, trust, confidentiality and satisfaction.

To enrich user experience, to rationalize the investment in cloud, achieving QoS targets is a must. Existing systems on QoS [1]- [6] have tried to offer guarantee in accordance with the SLA (Service Level Agreement). Efficient resource management through multiple VM's multiplexing has been analyzed in [7]. Still, the aim of fulfilling Service Level Agreement is a big obstacle due to unclear and dynamic features of network and IT resources in cloud platform.

In this paper, QOS aware self-adaptable VM Management System has been presented that will serve as a flexible and effective Management of the cloud service. The aim of the model is that QoS target is met by controlling number the requests so that cloud service does not get unstable. The system also makes sure that cost is controlled by optimizing IT Infrastructure. Here, concurrent input queues have created based on the same nature of the cloud services so that VM established for a demand can be reused by next jobs in same queue, which in turn reduces the time of creation and destroying VM's to certain extent. To start, the most critical VM is chosen using a priority scheduler and based on the priority, resources availability, newer VM's is created.

2. Background and Related Work

QoS most important term associated with cloud users, who anticipate that the service providers to provide them all the essential characteristics, and for cloud service providers, find the balance between QoS achievement and cost of Operation [8]. Effective resource management and scaling are the main constraints of cloud service that are achieved by virtualization, mainly used for localization of errors and manageability [9].

L. Bin et al. [10] et al. suggested a QoS aware system that deletes redundant server to save space and lessen the maintenance cost. Simulation results have shown that model can save disk space and reduce maintenance costs, while the ensuring QoS Constraints are met. But the model increases the cost on updating and variation of data is usually connected with data duplication.

P. Zhang et al. [11] have put forward a QoS management for mobile cloud computing based on Fuzzy Cognitive Map. However, the paper has not explained the total no of request that will be served at any time, the method of handling the tasks and how the system acts if it becomes congested.

Y. Xiao et al. [12] have proposed an effective reputation-oriented QoS providing method that can lowers the price of resources, while fulfilling QoS constraints. But they have not dealt with integrating security and confidentiality requirements.

A responsive algorithm for forceful Virtual Machine establishment of cloud applications is given by the authors of [13] in a upbeat style. Resource management based on local and global rules for energy management in cloud servers is described in [14]. The model speaks a lot about local policies in achieving power management, it doesn't convey lot about global policies in meeting QoS Constraints.

P. C. Hershey et al. [15] model explains how QoS observing, and reply for systems that provide computing service via a cloud. It identifies places inside the management domains where QoS parameters can be monitored and maintained. The model is really effecient but it was not used in federated clouds over real time.

M. Salam et al. [16] presented a QoS- oriented federated cloud computing structure where many individual cloud service enablers can collaborate effortlessly to offer scalable QoS guaranteed services. The main important work of the model is its QoS enhancement that can activate the on-demand resource providing across different providers, hence increasing QoS constraints and resources usage, remove Service Level Agreement defilements and improve Service Level Agreement ratification. Still Difficult services were not created using a various service from different cloud service enablers and there was no mechanism to check Denial of Service(DDos) attacks

In our model we have focused on Virtual Machine(VM) provision plans over the cloud, in view of strict SLA.

3. Proposed System

3.1. System Architecture

The Environments, the constraints and the notations of the various parameters are provided in the section below. System consists of collection of data centers denoted by 'DC'. There is n no of Physical servers for each data center denoted by 'PS'. All the servers have the same number of resources (applications software, Connectivity, servers and services). Collection of Application instance is denoted by AI and the collection of VMs is denoted by VA. The required number of VM for an application is based on the application type and the workload in relation to time. Tneq is the time required requested by the client which the service is needed. Tactual is the time of completing the task. To satisfy QOS, Tactual should be equal or less than Tneq. Based on the requirement the task needs to be grouped.

3.2. System Design

For governing the ambiguous activities of the network components and to synch ever changing workload of the cloud computing setting, a self-adapting VM management system for meeting QoS standard is offered below. Self-adaptable QoS aware VM management system is shown in fig 1. Number of queues are formed as input based on the type of service requested. The Management system has the following modules: 1) Entry Controller/ Requirement Analyzer, is the entry for the requested service. When the system is overloaded and QOS standards cannot be met for new tasks, no service level agreement is promised to the client so no fresh tasks are allowed. Job of the analyzer is to identify amount of cloud computing resource required and to direct the service to the necessary queue 2) Resource provider, analyzes the system readiness and instructs Entry Controller/ Requirement Analyzer in deciding the tasks to be allowed the system. 3) Virtual Machine Server, identifies the higher priority/ critical task in the queues and offers VM to that task, before creating a new VM the system waits for the tasks whose last request are not still Completed.

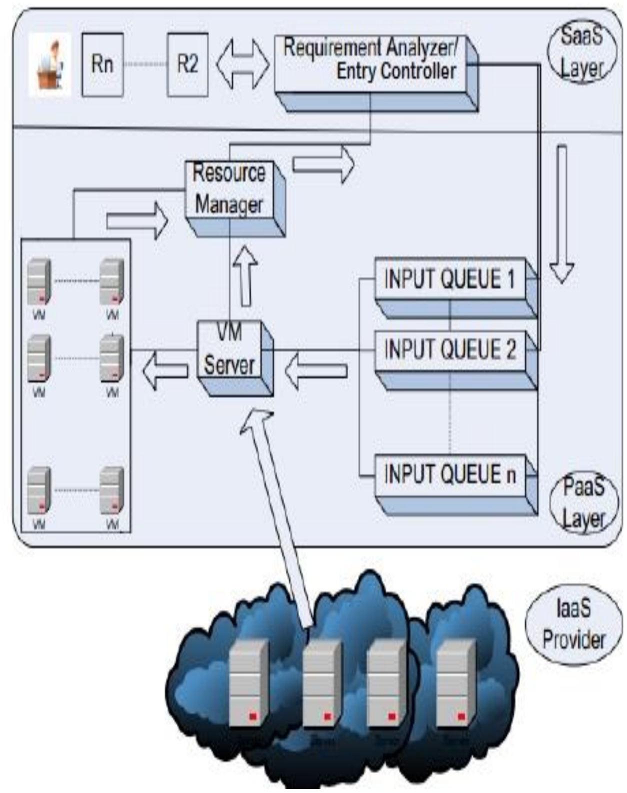

Fig. 1: Self-adaptable QoS aware VM management system

3.3. Self-Adaptable VM Management System

Figure 2 depicts the overview of the self-adaptable VM management Process. Entry controller/ requirement analyzer to start analyzes whether the resources available are adequate to serve the new task. If the task can be completed within the time as written in the SLA, it allows the task in to cloud computing system. Otherwise, it doesn't assure any guarantee of the service level agreement. The task is permitted to enter the queues based on the service required for the task. If no queue is available for such task a new queue is created. Due to the nature of requirement change due to time and also the predictor may be sometimes wrong, the virtual machine is reconfigured to match the size of the task if the change is within the prescribed value. When the size of the task is greater than the prescribed value, a new Virtual machine is created and the new and existing Virtual Machines are connected. The advantage of the above process is provisioning new VM is only needed for high time based priority tasks and the other task can be made reuse the VM provisioned for a high time based priority tasks that are same. VM Server identifies the tasks that are with high priority and also the virtual machine can be killed and the VM that are to be newly provisioned, so that all the tasks can be finished within the guaranteed time and high QoS standards can be sustained

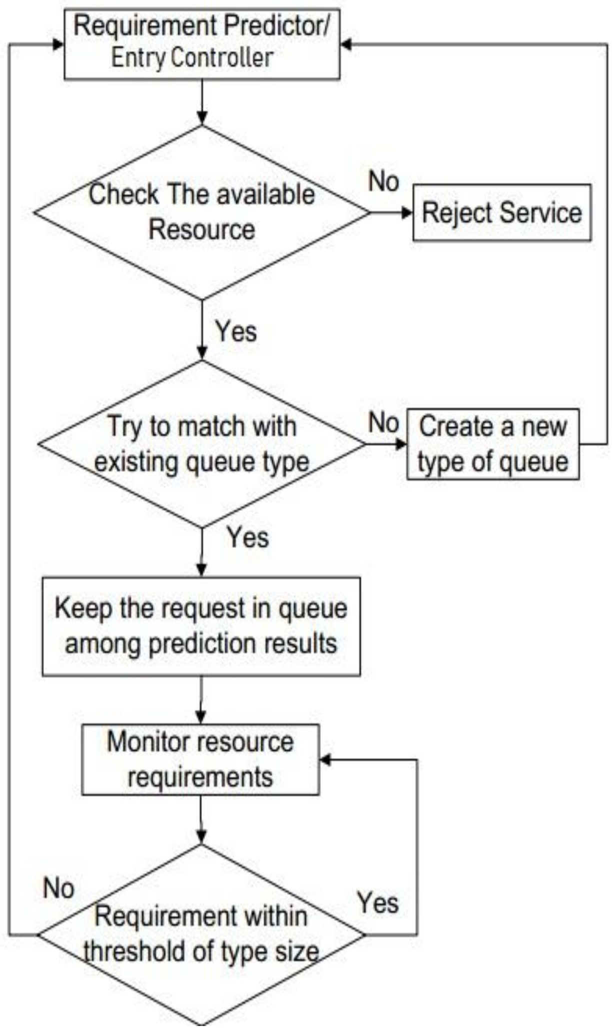

Fig. 2: Qos Aware self-adaptable VM management Processor

3.4. QOS Aware Self Adaptable VM Management Algorithm

QoS is provided allowing the tasks in to the system in controlled manner and carefully creating new Virtual Machine. Before permitting a task to be served, the sum of time allowed for all the task in the system and in the queue is summed and denoted by Tassured. Tavg is calculated as the average of the time taken to complete all the tasks in service and queue. The total system computational time is given by Timetotal and is summed up with the reserved value of time provide to sustain ever changing workload and uncertain tasks. Every time a new task arrives at entry controller/ Requirement analyzer, it's actual system working time Tnewact is predicted. If the reserved time Tnewneg and Tneg is greater than the sum of Timetotal, Tactual, reserved time and the whole affordable service time Taffservice is higher than the sum of Tneg and Tnewactuat, then the new task is given a slot in the queue for service.

The task enter the queue that matches its requirement of service. New VMs are created for some queues and the tasks of higher priority have the chance of completing its process earlier. As soon as the VM finishes the task allocated to it, the VM is reallocated with a new task so creation of new VM's are minimized Tasks are given service of VM's based on the time they enter the queue and priority of the job. The priority value is a calculated using the formula:

Priority value = arrival time + negotiated time

Virtual Machine server provides new VM for all the tasks waiting for entry in to the service that which has the maximum total priority value for all of its tasks.

Algorithm Self Adaptable Virtual Machine Management System
INPUT:
Tavg: Monitored average system time,
Tser: Total system time that can be provided by the Cloud controller,
ResTime: Time for Optimal QoS Support,
ta : No of tasks,
q : No of queues
tq : No of tasks in queue q

QOS aware self adaptable VM Management System
1. Tassured <- Σ Treq;
2. Timetotal <- n * Tavg;
3. Testimate <- Tnewact + Timetotal + ResTime;
4. Tmaxlimit<- Tneg + Tnewneg;
5. if Tmaxlimit >= Testimate&&Tser >= Tmaxlimit then
6. Allow task into the queue;
7. else
8. Reject task from entering the queue;
9. end if
10. Determine priority value;
11. if resources available for creating Virtual Machine then
12. Do for all jobs waiting for service;
13. Calculate priority _ Σ priorityfactori;
14. Find max priorityj for establishing new Virtual Machine of type j;
15. else
16. Hold for Virtual Machine of same type for finishing the existing task in the system;
17. end if

4. Performance Assessment

The simulation of the system is run CloudSim tool. Simulation is carried out by setting up data centers which consists of 100 system of quadcore processor and 8GB of DDRRAM. Rate of tasks entering is set at 500 tasks per sec. Time required to commission a Virtual Machine is 2-3 mins and the time for completing a task is considered as 30-50 mins. The results area correlation of Cloud performance between our QOS aware self-adaptable VM Management System and Traditional VM Provisioning system and Fig. 3. No of virtual machine vs Establishment Time (TVM). The graph show that huge amount of time is saved by establishing new Virtual Machine that can be used to cater extra tasks and also minimize the rejection count.

In fig 3, the correlation of time for VM establishing with no of VMs is shown. In the case where all the tasks are of the same nature so all the established Virtual Machines can be reused so that time for establishing VM's is reduced. In cases where tasks are of different nature and new Virtual Machine is needed for all the tasks. So performance of the system is identical to the traditional models. In case where all the tasks are of same type, time is needed to establish only the first Virtual Machine, all the other tasks can be served by reusing the existing Virtual Machine. In traditional Virtual Machines the time to serve a task is same for both best and worst case. Our model QSVM will behave like AVM in worst scenario.

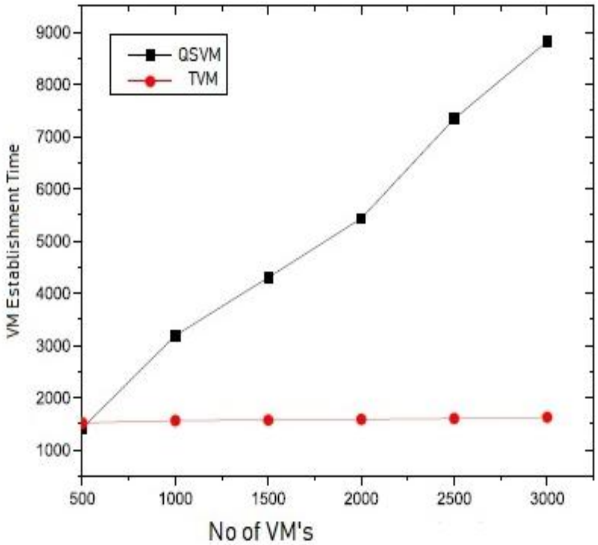

Fig. 3: No of VM, s vs VM's Establishment Time

Fig 4, show that the correlation between the QSVM and TVM shows that no of tasks served with respect to time. The time to server new tasks is comparatively less in our system because the time needed to establish new virtual machines is removed if the tasks is of the same type. Hence more request can be served at the less time there by increasing user satisfaction and also more revenue to service provider.

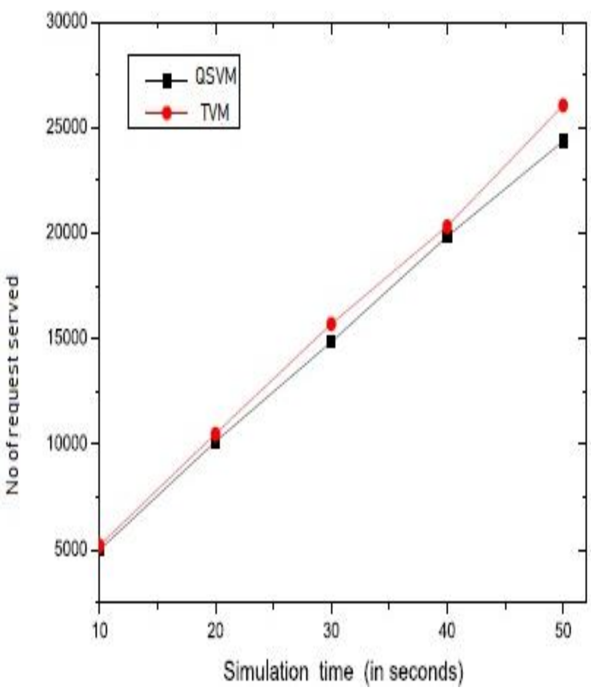

Fig. 4: No of Tasks Served vs Simulation Time

Fig 5 shows the rejection rate of task in given time. The correlation between QSVM and TVM shows QSVM has less no rejection because as the time to establish new Virtual machine is removed leading the time being used serve new tasks hence the no of rejection or less.so QSVM becomes more efficient during peak traffic duration.

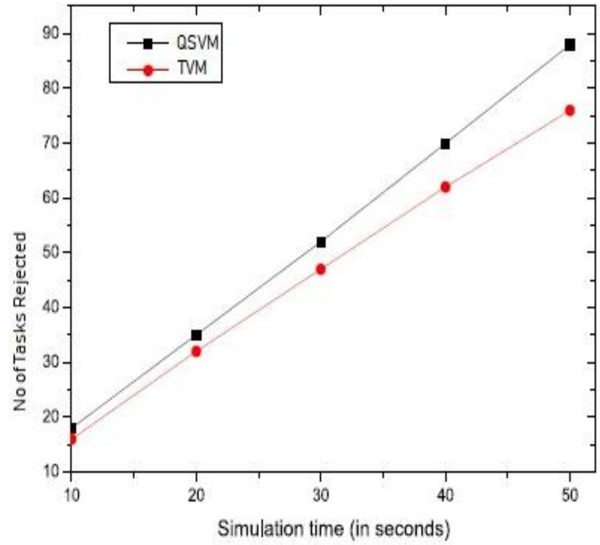

Fig. 5: No of Tasks Rejected Vs Simulation Time

5. Conclusion

Though Cloud computing provide lots of advantages, at the same time it also faces lot of obstacles like providing better user satisfaction within the written SLA. For eradicating this obstacles creation of New VMs in cloud computing we have presented here an QOS aware self-adaptable VM Management system. We have stated problem of establishing new VM frequently and provided a QoS aware system to overcome this problem. we have also provided an algorithm to lessen the no of rejection. The aim of the system is achieving QoS targets by reducing the no of rejection in clouds. Simulation results also shows that QSVM will provide a more stable performance in conjunction with QoS Criteria.

References

[1] G. Lodi, F. Panzer, D. Rossi and E. Turin, "SLA-Driven Clustering of QoS-Aware Application Servers," in IEEE Transactions on Software Engineering, VOL. 33, NO. 3, pp. 186-197, March 2007.

[2] V. Stanched and C. Schroder, "Negotiating and Enforcing QoS and SLAs in Grid and Cloud Computing," in GPC '09 Proceedings of the 4th International Conference on Advances in Grid and Pervasive Computing, November 2009.

[3] X. Wang, Z. Du, X. Liu, H. Xin, X. Jian, "An adaptive QoS management framework for Void cloud service centers. 2010 International Conference on Computer Application and System Modeling (ICCASM), Volume: 1, 2010, pp. 527-532.

[4] Y. Ye, N. Jain, L. Xia, S. Joshi, I-L. Yen, F. Bastani, K. L. Cureton, M. K. Bowler, "A Framework for QoS and Power Management in a Service Cloud Environment with Mobile Devices" in 2010 Fifth IEEE International Symposium on Service Oriented System Engineering (SOSE), pp. 236 - 243.

[5] Q. Li, Q. Hao, L. Xiao and Z. Li, "Adaptive Management of Virtualized Resources in Cloud Computing Using Feedback Control," in 2009 1st International Conference on Information Science and Engineering (ICISE), pp. 99 - 102, 2009.

[6] Y. Xiao, C. Lin, Y. Jiang, X Chu and X. Shen, "Reputation-Based QoS Provisioning in Cloud Computing via Dirichlet Multinomial Model," in 2010 IEEE International Conference on Communications (ICC), pp. 1 - 5, 2010.

[7] X. Meng, C. Isci, J. Kephart, L. Zhang and E. Bouillet, "Efficient Resource Provisioning in Compute Clouds via VM Multiplexing," in ICAC10, 2010. 467

[8] D. Ardagna, G. Casale, M. Ciavotta, J. F. Perez, W. Wang, “Quality-of-Service in cloud computing: modelling techniques and their applications”, Journal of Internet Services and Applications, Volume: 5, Issue: 11, pp.1-13, 2014.
[9] R. Nathuji, A. Kansal and A. Ghaffarkhah, “Q-Clouds: Managing Performance Interference Effects for QoS-Aware Clouds,” in EuroSys10, 2010.
[10] L. Bin, Y.Jiong, S. Hua, N. Mei, “A QoS-aware dynamic data replica deletion strategy for distributed storage systems under cloud computing environments”, in Proc. Second Int. Conf. on Cloud and Green Computing, pp. 219-225, 2012.
[11] P. Zhang and Z. Yan, “A QoS-AWARE SYSTEM FOR MOBILE CLOUD COMPUTING,” in Proceedings of IEEE CCIS2011, 2011.
[12] Y. Xiao, C. Lin, Y. Yiang, X. Chu, X. Shen, “Reputation-based QoS provisioning in cloud computing via Dirichletmultinomial model”, IEEE ICC Proceedings, pp. 1-5 2010.
[13] T. C. Chieu, A. Mohindra, A. A. Karve, and A. Segal, “Dynamic scaling of web applications in a virtualized cloud computing environment,” in Proceedings of the 6th International Conference on e-Business Engineering (ICEBE09), 2009.
[14] R. Nathuji and K. Schwan, “Virtual power: Coordinated power management in virtualized enterprise systems,” in ACM SIGOPS Operating Systems Review, vol. 41, no.6, pp. 265278, 2007.
[15] P. C. Hershey, S. Rao, C. B. Silio and A. Narayan, “System of systems for Quality-of-Service observation and response in cloud computing environment”, IEEE Systems Journal, Volume: 9,Issue:1, pp. 1-5, 2015.
[16] M. Salam and A. Shawish, “A QoS-oriented inter-cloud federation framework”, IEEE Systems Journal, pp. 642-643, 2015.