Multi-Objective Tasks Scheduling Algorithm for Cloud Computing Throughput Optimization

Atul Vikas Lakra\textsuperscript{a}, Dharmendra Kumar Yadav\textsuperscript{b}

\textsuperscript{a}Assistant Professor, Department of IT, VSSUT Burla, Sambalpur -768018, India
\textsuperscript{b}Associate professor, Department of CSE, MNNIT Allahabad-211004, India

Abstract
In cloud computing datacenter expert server unification to enhance the efficiency of resources. Many Vms (virtual machine) are running on each datacenter to utilize the resources efficiently. Most of the time cloud resources are underutilized due to poor scheduling of task (or application) in datacenter. In this paper, we propose a multi-objective task scheduling algorithm formappingtasks to a Vms in order to improve the throughput of the datacenter and reduce the cost without violating the SLA (Service Level Agreement) for an application in cloud SaaS environment. The proposed algorithm provides an optimal scheduling method. Most of the algorithms schedule tasks based on single criteria (i.e execution time). But in cloud environment it is required to consider various criteria like execution time, cost, bandwidth of user etc. This algorithm is simulated using CloudSim simulator and the result shows better performance and improved throughput.

1. INTRODUCTION
Within few years cloud computing grab the IT market very fast and most of the IT industry start using the cloud computing. In cloud computing the word cloud refers as internet, so the meaning of cloud computing is Internet Based Computing. In other words it’s a kind of server based computing. Cloud computing provide on demand services to the client. The services includes SaaS (Software as a service) where application software and database access provided to the user pay per use basis, IaaS (infrastructure as a Service) where virtual machine provided to the user using virtualization of physical machine which includes processing power, storage and other resources, PaaS (Platform as a Service) where cloud provider provides a computing platform which includes OS, programming language execution platform and web server. Cloud computing serves on demand requests of the users with self-managed virtual infra-structure and with efficient resources utilization. Growth of cloud computing slow down the efficiency, throughput and utilization of resources for which cloud computing need to be evolved. Apart from many ways to enhance the throughput and efficient resource utilization one way is the cloud task scheduling. Through task scheduling we can manage the resource utilization which in turn increases the throughput of the system. Scheduling refers to the mapping or assigning a task to a specific Vm, such that resource utilization increase. An efficient task scheduling algorithm improves the overall system performance and helpsservice provider to provide good quality of services (QoS). In cloud computing broker plays an important role. Brokers have the list of Vms and its QoS. A high performance Vm assign with the high QoS. Broker takes the requests from the user and sends the request.
to the one of the VM which meets the user requirement and the service level argument (SLA). At the time of assigning the request to a particular VM the quality of service (QoS) for the request or task should not decrease. Sometimes a good QoS task is assign to a low QoSVm which leads to the poor utilization of resources and this violates the SLA. So an efficient task scheduling algorithm should be implemented at the broker.

Rest of the paper is organized as follows. The section 2, describe some of the works related in the area of task scheduling. The section 3, describe the proposed work. The section 4, describe the experiment setup and simulation results. The section 5, conclude the proposed algorithm.

2. Related Work

Cloud computing is a new technology and still is in the developing stage. Cloud computing enhances its performance and throughput by using an efficient task scheduling algorithm. Most of the task scheduling algorithm, for cloud computing have been proposed in the last few years are based on QoS. QoS parameters include execution time, deadline, cost, bandwidth of communication; make span, reliability, scalability and many others. Based on QoS parameters a task is selected for the execution on a selected VM, which increases the resource utilization and the throughput of the overall system.

One of the traditional methods for selecting a task from a group of tasks has been done by priority scheduling. Priority of a task can be assigned dynamically using the QoS parameter at runtime. Static priority assignment for tasks faces many difficulties. QoS has the ability to provide different priority to different applications, users, or data flows, or to guarantee a certain level performance data flow. In [6, 10] authors assigns priority to different tasks by taking QoS parameter such as execution time and cost of application and QoS value.

In [1, 11] authors proposed an algorithm for task scheduling based on multiple criteria and multiple decision to choose a task to be executed in a particular VM. Multiple criteria include the various QoS parameters. These algorithm helps to reduce the make span of the system.

Optimized task scheduling algorithms using genetic algorithms put system into an optimal stage without trapping the system to a local optimal stage. In [5] authors proposed an algorithm based on NSGA-II for load balancing of CPU, memory and bandwidth in cloud computing and [4] author uses the combination of genetic algorithm along with fuzzy optimization theory. Nature inspired algorithm is also used, such as ant colony optimization. In ant colony optimization ant moves in random direction for the search of food source around the colony. Here the ants are tasks and the food sources are VMs. In [14, 15] authors implements modified ant colony optimization to minimizing the execution time and cost by considering execution time, arrival time and other QoS parameter as a criteria for searching a best VM for the execution of tasks such that the make span of the system is reduced.

The main job of broker is to allocate the VM to a task. At runtime broker decides mapping of task to a VM. Sometimes single tasks with multiple users [3] are mapped to VM and sometimes from a group of task a particular task is picked up for the allocation of VM depending upon the execution time and arrival time [6, 7, 9, 12, 16]. The tasks in the group is selected sequentially and submitted to the Virtual Machine. The process of allocation is done repeatedly until the entire tasks in the queue finish its execution. This leads to a minimized make span of the VMs and reduces completion time or execution time of task. Main goal of all tasks scheduling algorithm is to minimize the execution time, cost, make span but few algorithm has been proposed to increase the scalability [13] and reliability [15] of the whole system. These task scheduling algorithms increase the QoS of the system.

3. Proposed Work

3.1 Introduction

Cloud computing service providers have several datacentres in order to optimally serve customer needs around the world. However, existing system does not provide the proper scheduling of customer requested application among the VMs in datacenters to achieve reasonable QoS levels. Every datacenter in cloud computing consist of numerous servers and each server runs numerous VMs. Each VMs have different capability to execute different QoS’s tasks requested by the customer.
Atul Vikas Lakra and Dharmendra Kumar Yadav / Procedia Computer Science 48 (2015) 107 – 113

The cloud computing architecture and the proposed work is shown in Fig. 1. Cloud broker is responsible for mediating negotiations between SaaS and cloud provider and such negotiation are driven by QoS requirements. Broker acts on behalf of SaaS for allocation of resources that can meet application’s QoS requirements.

3.2 Problem Statement

The main problem is to bind set of tasks received by the broker to the received list of VMs, so that execution time of workload is reduced to minimal optimized time. Single objective scheduling algorithms have some problem. In priority task scheduling [22, 23], high priority tasks always get chance to execute, due to this low priority task have to wait for a long time. Sometimes low priority task gets a chance to execute but, if high priority tasks keep coming then low priority task is preempted and CPU is allocated to high priority task and this leads to increase in execution time of a task as well as it reduces the throughput of the system. Similarly in First Come First Serve [22, 23] and Shortest Job First task scheduling [22, 23] algorithms face problem in worst case scenario. These algorithms perform very well in the best case but in worst case they degrade the performance to very low level. So an efficient scheduling algorithm is required which can provide optimized performance in both cases. Using a proper scheduling algorithm implementation in broker improves the datacenter’s performance without violating SLA. The order of task submission and the VMs also influence the execution time of the entire workload.

VM characteristic define as $V_i = \{ID_i, M_i\}$, where $V_i$ is the $i$th Virtual machine, $ID_i$ is the ID of VM and $M_i$ is the MIPS of $i$th VM. Task characteristic define as $T_i = \{ID_i, Q_i, S_i\}$, Where $ID_i$ is the task ID, $Q_i$ is the QoS value of a task and $S_i$ is the size of the task(MI) of $i$th Task.

3.3 Assigning QoS for Tasks and VMs

Cloud broker sends request to the cloud service provider for the QoS of requested task. In proposed task scheduling algorithm task’s priority is assigned according to the QoS. High QoS task assigned with low QoS value and the low QoS task assigned with high QoS value. Hence, the task with lower QoS value is a high priority and the task with high QoS value is a low priority. The QoS for tasks are documented in SLA. Task’s QoS value is associated
throughout its life cycle. Also cloud broker sends request to the cloud service provider for list of VMs created in the
datacenters. After receiving the list of VMs cloud broker assigns QoS to the VMs. Millions of instructions per
second (MIPS) of a VM is considered for assigning VM’s QoS. VM with high MIPS is a high QoS VM and VM
with low MIPS is low QoS VM.

3.4 Non-dominated sorting

A non-dominated sorting [8, 5] is used to solve the multi-objective problems. In multi-objective problems multiple
objective functions are considered. In the proposed work, the main goal is to minimize the execution time of a task.
Main goal is achieved by selecting a task with minimum task size and minimum (low)QoS value. The two objective
functions are as follows.

Minimize \( f(S_i) = S_i \) \( \forall j \exists i, f(S_i) \leq f(S_j) \) \hspace{1cm} (1)

Minimize \( f(Q_k) = Q_k \) \( \forall j \exists i, f(Q_i) \leq f(Q_j) \) \hspace{1cm} (2)

A non-dominated sorting is used to implement the multi-objective task scheduling algorithm with the above
objectives. In non-dominated sorting, multiple objectives are applied at a time.

3.5 Virtual Machine Selection

Cloud broker maintains a list of VM for VMs. This list is updated in fixed timeinterval and dynamically at the
pick time when number of requests increase suddenly.According to the MIPS of VM, list of VMs is sorted in
descending order. VM in the first position of the list have high QoS and at the end of the list low QoS VM. After
non-dominated sorting finally generated non-dominated task’s set is bound with the VMs. In the process of binding or
allocation of VM to a task, it is done sequentially according to both the list of tasks and VMs. The first VM from the
VM’s list to the first task in the task’s list and second VM in the VM’s list with second task in the task’s list. Once the
allocation reached the last VM, the next task will be submitted once again to the first VM of the VM’s list and the
process of allocation will be repeated for all tasks. The proposed multi-objective task scheduling algorithm
is described in Algorithm 1.
Algorithm 1 : Multi-objective task scheduling algorithm

1. Submit both VMs list of successfully created VMs in datacenter and task list to Broker.
2. Create a received list of tasks.
3. Create a received list of VMs.
4. Non-dominated sorting (list of task)
   \[ i \leftarrow 0 \]
   Create empty non-dominated list
   dominated list ← list of task
   Initially put task\(_i\) in the non-dominated list
   for all \( i \leftarrow 1 \) to size of task’s list do
     for all \( j \leftarrow 0 \) to size of non-dominated list do
       if task\(_j\) dominates task\(_i\) then
         put task\(_j\) into non dominated set
       else
         if task\(_i\) dominates task\(_j\) then
           put task\(_i\) into non dominated set
         end if
       else
         put task\(_i\) and task\(_j\) into non dominated set
       end if
     end for
   end for
5. Sort the list of task according to the non-dominated task set.
6. Sort the VM received list in descending order.
7. \( j \leftarrow 0 \)
   for all \( i \leftarrow 0 \) to the size of task’s list do
     if \( j \geq 0 \) then
       Bind task\(_i\) to the VM\(_j\) j++
     if j == number of VMs then
       j=0
     end if
   end if
end for

4. Experiment & Results

In order to obtain results of the proposed algorithm the simulation was done using CloudSim 3.0.2 simulator on windows 7 OS with Core i3 2.10GHz processor. NetBeans IDE 7.2.1 is used to run CloudSim 3.0.2. In our simulation scenario, the proposed algorithm is compared to the existing task scheduling algorithm, for this purpose following illustrative example is taken. We have created many VMs and tasks with different task size. Task size ranges from 1000 to 8000 and the QoS valueranges from 0-9. Task which has low QoS value has the higher priority. The VMs have been created which have processing power ranges from 1000-5000MIPS.

Table 1. Workload
The comparison of turnaround time is shown in Fig. 2 of three task scheduling algorithms with six different workloads as tabulated in Table 1. From the comparison of these three tasks scheduling algorithms it is observed that the proposed algorithm performs better than other two algorithms with minimum execution time and increased throughput of the cloud computing system.

5. Conclusion

In proposed multi-objective task scheduling algorithm for cloud computing environment is an optimal task scheduling algorithm which provides the minimum overall execution time. Cloud computing works in real time and single criteria based algorithm may not be the one for task scheduling. The proposed algorithm can be improved by taking consideration of some other QoS parameters.

References

1. Shamsollah Ghanbari, Mohamed Othman “A Priority based Job Scheduling Algorithm in Cloud Computing”, International Conference on Advances Science and Contemporary Engineering 2012 (ICASCE 2012).
2. El-Sayed T. El-kenawy, Ali Ibrahim El-Desoky, Mohamed F. Al-rahamawy “Extended Max-Min Scheduling Using Petri Net and Load Balancing,” International Journal of Soft Computing and Engineering (IJSCCE) ISSN:2231-2307, Volume-2, Issue-4, September 2012.
3. Shalmali Ambike, Dipti Bhansali, Jaeekshirsagar, Juhli Banswal “An Optimistic Differentiated Job Scheduling System for Cloud Computing,” International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 2, March-Apr 2012, pp.1212-1214.
4. Sandeep Tayal “Task Scheduling optimization for the Cloud Computing Systems”, “IJEST” International journal of advanced engineering sciences and technologies Vol No. 5, Issue No. 2, 111-115.
5. Jianfeng Zhao; Wenhua Zeng; Min Liu; Guangming Li; Min Liu, “Multiobjective optimization model of virtual resources scheduling under cloud computing and it’s solution,” Cloud and Service Computing (CSC), 2011 International Conference on , vol., no., pp.185,190, 12-14 Dec. 2011.
6. Wei-Jen Wang, Yue-Shan Chang*, Win-Tsun Lo, and Yi-Kang Lee, “Adaptive Scheduling for Parallel Tasks with QoS Satisfaction for Hybrid Cloud Environments,” Journal of Supercomputing, DOI: 10.1007/s11227-013-0890-2. (SCI) 2013.

7. Achar, R.; Thilagam, P.S.; Shwetha, D.; Pooja, H.; Roshni; Andrea, “Optimal scheduling of computational task in cloud using Virtual Machine Tree,” Emerging Applications of Information Technology (EAIT), 2012 Third International Conference on, vol., no., pp.143,146, Nov. 30 2012-Dec. 1 2012.

8. “Non-Dominated sorting” en.wikipedia.org/wiki/Multi-objective optimization.

9. Das, A.K.; Adhikary, T.; Razzaque, M.A.; ChoongSeon Hong, “An intelligent approach for virtual machine and QoS provisioning in cloud Computing,” Information Networking (ICOIN), 2013 International Conference on, vol., no., pp.462,467, 28-30 Jan. 2013.

10. Moses, J.; Iyer, R.; Illikkal, R.; Srinivasan, S.; Asiospos, K., “Shared Resource Monitoring and Throughput Optimization in Cloud-Computing Datacenters,” Parallel & Distributed Processing Symposium (IPDPS), 2011 IEEE International, vol., no., pp.1024,1033, 16-20 May 2011.

11. Hilda Lawrence, Dr. Salaja Silas “Efficient QoS Based Resource Scheduling Using PAPRIKA Method for Cloud Computing,” International Journal of Engineering Science and Technology (IJEST) Vol. 5, No.03 pp 638-643 March 2013.

12. C. Lin, S. Lu, “Scheduling Scientific Workflows Elastically for Cloud Computing,” in IEEE 4th International Conference on Cloud Computing, pp. 246-247, 2011.

13. Mrs. S. Selvarani; Dr. G. Sudha Sadhasivam, “Improved cost-based algorithm for task scheduling in Cloud computing,” IEEE 2010.

14. Raju, R.; Babukarthik, R.G.; Chandramohan, D.; Dhavachelvan, P.; Vengattaraman, T., “Minimizing the make span using Hybrid algorithm for cloud computing,” Advance Computing Conference (IACC), 2013 IEEE 3rd International, vol., no., pp.957,962, 22-23 Feb. 2013.

15. R. Gogulan, A. Kavitha, U. Karthick Kumar “An Multiple Pheromone Algorithm for Cloud Scheduling With Various QoS Requirements,” IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 3, No 1, May 2012.

16. Nguyen, Quyet Thang; Quang-Hung, Nguyen; Tuong, Nguyen Huynh; Tran, Van Hoi; Thoai, Nam, “Virtual machine allocation in cloud computing for minimizing total execution time on each machine,” Computing, Management and Telecommunications (ComManTel), 2013 International Conference on, vol., no., pp.241,245, 21-24 Jan. 2013.

17. Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose, and Rajkumar Buyya, “CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms,” Software: Practice and Experience (SPE), Volume 41, Number 1, Pages: 23-50, ISSN: 0038-0644, Wiley Press, New York, USA, January, 2011.

18. I. Foster, Y. Zhao, I. Raiciu, and S. Lu. “Cloud computing and grid computing 360-degree compared.” In Proceedings of Grid Computing Environments Workshop, pages 1—10, 2008.

19. D. M. S. Daryl C. Plummer, David W. Cearley. “Cloud computing confusion leads to opportunity. Technical report,” Gartner Research, 2008.

20. P. Mell and T. Grance. The NIST Definition of Cloud Computing (Draft). National Institute of Standards and Technology, 53:7, 2010.

21. Cloud computing and distributed computing. http://www.cncloudcomputing.co

22. Tasks Scheduling (computing) https://en.wikipedia.org/wiki/Scheduling (computing).

23. Tasks Scheduling http://www.personal.kent.edu/ rmuhamma/OpSystems/os.html