Budget and Time Based Efficient Task Scheduling on Cloud

Amit Kumar, Arushi Kumar
University of Pune
Army Institute of Technology
Pune, India

Narendra Gehlot , Sourav Kumar
University of Pune
Army Institute of Technology
Pune,India

***Abstract*- Scheduling in Cloud Computing has been a major challenge. Cloud Computing is a technology that is achieved by Distributed, Parallel and Grid Computing. Scheduling on cloud is done on the basis of various factors including execution time, response time, flow time, reliability, load balancing, etc. What we introduce in this paper is a scheduling algorithm using a combination of completion time of a job on the cloud and budget given by the user. This algorithm reduces the estimated budget given by the user by utilizing the available resources efficiently according to the type of task submitted.**

Keywords – Scheduling ; Budget ; Cloud Computing ;Load Balancing; Job Scheduling

I. INTRODUCTION

Cloud Computing is a type of computing that is based on sharing computing resources rather than having local servers to handle the applications. “Cloud” basically refers to “the internet” so, cloud computing would be “a type of computing on the internet” where different services such as infrastructure, software, database, platform, etc. can be provided to the organization’s computer on the backbone of the internet. Cloud Computing is comparable to grid computing , a type of computing where unused processing cycles of all computers in a network are harnessed to solve problems too intensive for any stand-alone machine. With the growing demand of the Cloud based services, efficient scheduling on cloud has now become of paramount importance. Scheduling refers to the set of policies to control the order of work to be performed by a computer system. There have been various types of scheduling algorithm existing in distributed computing system, and job
Scheduling is one of them. The main advantage of job scheduling algorithm is to achieve high performance computing and the best system throughput. Scheduling manages availability of CPU memory and good scheduling policy gives maximum utilization of resource.
Efficient execution of any scheduling algorithm is based on various parameters and various scheduling algorithms utilizing a specific combination of these parameters, are available to do the scheduling on cloud. Out of all the combinations of parameters budget, time, heterogeneity and load balancing have never been considered together. In this paper we propose an algorithm which not only utilizes all the resources efficiently but also gives the output in minimum budget. Such type of scheduling would be highly beneficial to the user as the job is performed on the cloud considering the deadline and the budgetary constraints.

A. Scheduling on Cloud

There are a lot of scheduling algorithms developed so far. These algorithms have considered various parameters to schedule the jobs on cloud. These parameters include make span, acceptance rate, completion time, priority of job, fairness, etc.

Task scheduling process is an allocation of one or more time intervals to one or more resources .In cloud computing, there is a problem of scheduling a set of submitted tasks from different users on a set of computing resources to minimize the completion time of a specific task or the makespan of a system. There are many other parameters that can be mentioned as factors of scheduling problem to be considered such as load balancing, system throughput, service reliability, service cost, system utilization and so forth. Through comprehensive study of scheduling, Task scheduling algorithm is a decision making process about assigning and finding the best match between tasks and resources. So scheduling **is NP-complete problem.**
For producing a scheduler, assume that we have m Resources Rj(R1, R2, ..,Rm) and we process n tasks Ti(T1, T2, .., Tn) to be mapped on these resources. Also expected execution time Eij of task Ti on resource Rj is defined as required time of resource Rj to finish task Ti provided that Rj has no load when assignment occurs. On the other side, expected completion time Cij of task Ti on resource Rj is defined as the overall time
Consumption till finishing any assigned task previously assigned. Assume Ri denote the beginning of execution task Ti.
From previous mentions, it can be concluded that

$$Cij = Ri + Eij$$

The makespan of complete schedule is defined as Max (Ci) where Ci is the completion time for a task Ti .
Makespan is defined as a measure of the throughput of the heterogeneous computing system; like the Cloud Computing environment
Scheduling algorithms can be categorized according to many polices as immediate and batch scheduling, preemptive and Non-preemptive scheduling, static and dynamic scheduling, etc

In Immediate mode, tasks are scheduled as soon as arrive the computing environment, while in the batch mode, tasks are grouped into a batch; that is a set of meta-tasks would be allocated at times called mapping events

For example, in the Minimum Execution time (MET) algorithm estimating the execution time of the submitted tasks on the available resources is calculated choosing each task to a resource would produce the minimum execution time for that task.

A systematic comparison of all the algorithms has been given in **table 1.1** below. Observing the table we realize the parameters cost and completion time have not been considered together.

II. Algorithm Proposed

The algorithm we propose utilizes the positives of min-min algorithm and preemptive round robin. Here, user provides the minimum requirements needed for deciding the priority of the job while submitting the task. The job is submitted in the form of subtasks assuming the fact that these sub tasks are independent of each other. These requirements include expected deadline for the job, the estimated budget he has and the type of job he wants to execute. We used the simulator *cloudsim* (http://www.cloudbus.org/cloudsim/) to implement the algorithm and compare the results with other algorithms after a few modifications on the simulator according to the requirements. So, when the requirements are submitted on cloud, a matrix is created. This is a P(*time* x *budget) matrix w*here time is as per the users requirements and the budget is calculated according to pre decided Billing time unit(BTU). The value is set on the basis of what kind of job would be executed on what kind of instance. For Example, if an arithmetic job is to be executed on an instance which has better resources for an arithmetic job its budget would be less than if the job is executed on an instance with better requirements for networking. We aim at providing right resources for the right job(heterogeneity factor has been factored in), thereby enhancing the efficiency , thus proving an optimal, cost effective and efficient solution. Hence, the budget would be varied according to job and instance type utilizing the resources in an optimized manner. In This matrix, we check the minimum value of P(*time* x *budget*) for the job on all the available instances and selects the minimum value. When the first job is submitted the completion time is same as the execution time but for other jobs the completion time has to be calculated keeping in mind the completion time of all the jobs already being executed on the instances. So after the minimum P value is selected for the job i.e. a particular instance is selected for the job, this job is assigned to that selected Instance. The job is executed on the VM in round robin manner with a pre decided time slice. For the first job there is no execution and calculation problem but for the second job and the jobs after that the VMs are already busy with a particular job running on each of them. The problems at this stage can be the following:

1. The Already running job's estimated deadline would increase since in round robin more number of jobs will get accommodated.
2. With increase in completion time the selection of more efficient instance, as we selected using min-min algorithm, would also get affected since the increase in time would lead to increase in the P value which might make the P value of some other instance lesser than the P value previously selected after applying the min-min algorithm.

To deal with this problem a regular updating of the initially created (time x budget) matrix is needed. For this, another matrix is associated with each instance. This keeps a record of all the cycles already performed on the instance, the jobs running on it, and the cycle during which the jobs started their execution. As soon as the instance Is initialized , the table Is created, and a counter keeps count of the cycles on the VM. When a new job is assigned to it, it is noted in this matrix , after updating all the jobs before it with the left number of cycles for those jobs, the execution starts in round robin manner. During the updating , the completion time of all the jobs is calculated considering the new job being assigned to the VM. So, the P matrix is updated by changing the completion time of all the jobs and again calculating the P factor. This changes can be calculated a follows:

$$New\ C_{ij} = (Cyc + N*(Cyc-1))*T_{ij}$$

Where,

Cij : completion time of the job

Cyc : Number of cycles for that job

N : number of cycles currently being executed on that VM

Tij : Time Slice

Which brings about two possibilities:

1. An increase in the completion time of the job might increase the deadline
2. This increase might not affect the deadline.

In both the cases we need the optimum solution so, after updating the central P matrix, we again apply min-min algorithm. If the new instance selected is the same as the previously selected instance the execution of the job continues on that instance else it is stopped and preemptedly executed on the selected instance with better value.

A. *Related Algorithms*

- Min-Min Algorithm:
- Round Robin:

B. *Simulations and Implementations*

Suppose user submit one job consisted of several subtasks then initially a 2D matrix is get created according to the execution time ,budget and type of job.

(Completion Time * Budget)

	M0	M1	M2	M3
ST0	200	250	220	300
ST1	150	170	190	160
ST2	300	320	180	360

After first step ST0 will get submitted to M0 machine according to max-min algorithm. Now we will update matrix by modifying the entries for M0 machine according to round robin (time slice=10).

(Completion Time * Budget)

	M0	M1	M2	M3
ST1	300	170	190	160
ST2	500	320	180	360

After second step ST2 will get submitted to M2 machine according to max-min algorithm. Again we will update matrix by modifying the entries for M2 machine according to round robin (time slice=10).

(Completion Time * Budget)

	M0	M1	M2	M3
ST1	300	170	370	160

After third step ST1 will get submitted to M1 machine according to max-min algorithm. Similarly further jobs will be get submitted considering both max-min and round-robin.

CONCLUSION AND FUTURE SCOPE

Here , we have proposed a new generic algorithm which is not only cost effective but also manages the resources on the basis of different job types. The instance allocation is done on the basis of different type of job. In the future as the type of job vary and increase, these types can be incorporated in the algorithm.

REFERENCES

[1] Vasileious Thanasias ,Choonhwa Lee and Sumi Helal ,"Budget Aware task Scheduling in the Cloud",IEEE,2014

[2] [1] 2.Zhao C., Zhang S., Liu Q., Xie J., Hu J., "Independent T asks Scheduling Based on Genetic Algorithm in Cloud Computing", IEEE 5th International Conference on Wireless Communications, Networking and Mobile Computing WiCom '09, Beijing, pp.1-4, 2009

[3] Huang Q.Y., Huang T.L.,"An Optimistic Job Scheduling Strategy based on QoS for Cloud Computing", IEEE International Conference on Intelligent Computing and Integrated Systems (ICISS), 2010, Guilin,
pp. 673-675, 2010R

[4] R. Buyya, C.S. Yeo, S.Venugopal, J. Broberg, I.Brandic .—Cloud computing and emerging IT platforms: vision ,hypeand reality for delivering computing as the 5th utility Future Generation Computing System, 2009, 25(6),p.599-616.

[5] Rajkumar Buyya1, Rajiv Ranjan2 and Rodrigo N.Calheiros, "Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit:Challenges and Opportunities".

[6] Sandeep Tayal, "Tasks Scheduling Optimization for thecloud computing systems", (IJAEST) International Journal Of Advanced Engineering Sciences and Technologies, Volume-5, No.2, p 11 – 15, 2011.

[7] R. Santhosh, T. Ravichandran, "Pre-emptive Scheduling of On-line Real Time Services with Task Migration for Cloud Computing", International Conference on Pattern Recognition,IEEE, Feburary 2013.