Review of resource scheduling in cloud computing

Sumit Dhaka[1], Mrs. Radhika Garg[2]

[1]Dept. of CSE, Vaish College of Engineering, Rohtak
[2]Assistant Professor, Vaish college of Engineering, Rohtak

Abstract: Cloud computing is known as service that allows use of computing, software, platform infrastructure through internet. In Cloud computing user pay for the use of these computing, software etc. In such environment there are plenty of resources at data center but with distributed nature of cloud and less information sharing between client and service provider present task scheduling algorithms become very important to provide efficient utilization of resources and in time delivery of output.

Keyword: Max-min algorithm, min-min Algorithm, RASA, task scheduling, Makespan.

INTRODUCTION

Cloud computing is known as service that allows use of computing, software, platform infrastructure on payment basic through internet. In today`s world cloud computing is growing at a rapid rate. It has also enabled to easy access to application and associated data from anywhere around the world. The main purpose of using this technology is to minimize cost and to maximize performance and efficacy. Preparation, timing, and failure management are required to implement the management, scheduling, and responding to demands in minimum time.

The main objective of this review paper is to study various scheduling policies for cloud environment and to examine their behavior with respect to deadline. Task scheduling is the basic requirement to make a number of cloud services for an efficient provider infrastruce while satisfying the client. Task scheduling algorithm is responsible for mapping jobs submitted to cloud environment onto available resource in such a way that total response time and make span is minimized along with customer satisfaction.

Many task scheduling algorithms are applied by resources manager in distributed computing to optimally allocate resources to tasks [11],[12].While other scheduling algorithms try to minimize the total completion time. Where the minimization is not necessarily related to the execution time of each single task, but also for minimize overall the completion time of all tasks. Job scheduling is one of the major activities performed in all the computing environments. Cloud computing is one the upcoming latest technology which is developing drastically. To efficiently increase the working of cloud computing environments, job scheduling is one the tasks performed in order to gain maximum profit.

Cloud Computing Characteristics

Cloud computing is almost taking advantage of the features that other computing and distributed grids own, but the proper use of the features has made this network superior to the other ones. Cloud computing owns six main characteristics as the following (Mittal and Soni, 2013):

Broad Access To Network

Access to cloud resources is possible throughout the network and standard methods are used for the users to access the network.

Supplying Service Based On Demand

Users can have access to their required resources and software without having to interact with cloud computing service providers.

Calculating Service (Pay Per Usage)

One of the key characteristics of cloud computing is calculating system based on the use of services and resources.

D**ensity Of Resources**

There are massive amounts of resources in cloud computing which are independent of their physical location via virtualization.

Multiple users (tenants) (shared resources)

It makes centralizing, increasing the use of unused resources, and sharing resources possible for the users.

Rapid Expansion Ability

Resources in the cloud should be able to expand rapidly and should be unlimited and accessible at any time form the users' point of view.

SCHEDULING

Job scheduling is an important task in cloud environment. Job Scheduling is used to allocate certain jobs to particular resources in particular time. In cloud computing, job- scheduling problem is a biggest and challenging issue. The main aim of job scheduling algorithm is to improve the performance and quality of service and at the same time maintaining the efficiency and fairness among the jobs and reduce the execution cost. An efficient job scheduling strategy must aim to yield less response time so that the execution of submitted jobs takes place within a possible minimum time. There are various scheduling strategies which should take care of all these things. But no such strategy exists which is concerned with both the users point of view as well as service providers point of view. In todays where client satisfaction is prime objective it is highly required that more information sharing exist between client and service provider , but such procedure are nt found in today’s scenario. The various job scheduling algorithms are the following.

A. First Come First Serve Algorithm: jobs are served in the order in which they arrive i.e jobs are queued and served in fifo format. This is simple and very quick algorithm but doesn’t provide much efficiency to job and resource optimization

B. Round Robin algorithm: In the round robin scheduling, processes are given a limited amount of CPU time called a time-slice or a quantum in FIFO manner. If a process does not complete execution before its CPU-time expires, the CPU is pre-empted and given to the next process waiting in a queue. And the preempted process is placed at the end of the ready queue and processed in the next time slice or quantum. In order to schedule processes fairly, a round-robin scheduler generally employs time-sharing, giving each job a time slot or quantam (its allowance of CPU time), and interrupting the job if it is not completed by then. The job is resumed next time a time slot is assigned to that process. If the process terminates or changes its state to waiting during its attributed time quantum, the scheduler selects the first process in the ready queue to execute. In the absence of time-sharing, or if the quanta were large relative to the sizes of the jobs, a process that produced large jobs would be favoured over other processes.

C. Min–Min algorithm: A type of algorithm in which short jobs execute in parallel and then followed by long jobs. The Min-min heuristic begins with the set of all unmapped tasks (Set U). Then, the set of minimum completion times, M, for each task ti ϵ U, is found. More over the task with the overall minimum completion time from M is selected and assigned to the corresponding machine (Thus named Min-min). At Last, the newly mapped task is deleted from U, and the process repeats until all tasks are mapped (i.e., U is empty).[13] However, Min-min considers all unmapped tasks during each mapping decision and MCT only considers one task at a time. Min-min maps the tasks in the order that changes the machine availability status by the least amount that any assignment could. Let ti be the first task mapped by Min-min onto an empty system. The machine that finishes ti the earliest, say mj , is also the machine that executes ti the fastest. For every task that Min-min maps after ti , the Min-min heuristic changes the availability status of mj by the least possible amount for every assignment. Therefore, the percentage of tasks assigned to their first choice (on the basis of execution time) is likely to be higher for Min-min than for Max-min (defined next). The expectation is that a smaller makespan can be obtained if more tasks are assigned to the machines that complete them the earliest and also execute them the fastest. Hear the main problem is regarding as Min-min gives higher priority to small tasks, it increases Response time for large tasks.

D. Max – Min algorithm: In reference to the heuristic which we have seen previously i.e. .Min-min Algorithm, the Max-min heuristic is also very similar to it. The Max-min heuristic begins with the set of unmapped tasks. Let U be the set of all unmapped tasks. Then, the set of minimum completion times, M, is found. The heuristic then selects overall maximum completion time from M and assigns to the resembling machine. And so it is named as Max-min Algorithm. Finally, the new mapped task is removed from U, and the process keeps on repeating until all tasks are mapped which implies till U is empty.[8] Intuitively, Max-min tries to minimize the penalties incurred from performing tasks with longer execution times. For instance, let the meta-task being mapped has many tasks. Assume that one task has short execution time and one has a very long execution time. Mapping the task with the longer execution time to its best machine first allows this task to be executed concurrently with the remaining tasks (with shorter execution times). For this case, this would be a better mapping technique than a Min-min mapping. Because in Min-min mapping all of the shortest jobs (tasks) would execute first, and then the longer running task would execute while several machines sit idle. Thus, in cases similar to this example, the Max-min heuristic improves makespan. It may give a mapping with a more balanced load across machines

E. RASA algorithm: This algorithm is the combination of max-min and min- min algorithm to perform in terms of both makespan and load balancing. In this algorithm max-min and min-min algorithm both are applied alternatively. Hence increasing efficiency. But even these algorithm doesn’t consider client requirement of completing task within a time period till which task is has utility for the client. Thus greater information sharing regarding task needs to shared between client and service provider.

CONCLUSION

In cloud computing environment, there are plenty of resources at datacenter which are needed to be allocated in efficient and effective manner to get the shortest response time, minimal possible completion time, and utilization of resources. To achieve all these factors, proper allocation of resources is the key element. There are various existing heuristic algorithms for resource scheduling and allocation in cloud computing. Min –Min and Max-Min are mainly used algorithms, but both of these have their own scope of better resource utilization.

RASA is a new algorithm which uses Max-min and Min-min algorithms. The algorithm determines to select one of these two algorithms, dependent on the standard deviation of the expected completion times of the tasks on each of the resources. RASA algorithm based on static and dynamic environment. But even the use of RASA doesn’t consider the deadline sensitivity of the jobs given to the cloud. All of the runnings algorithms are an attempt to get the possibly minimal completion time, and the best response time, but none of already existing techniques consider the deadline sensitivity of the job. The jobs to be executed over the cloud can be having different deadlines. Some might be very crucial to be completed before some other task. Some may lose their importance of execution if these are not completed in given time frame.

REFRENCES

[1] Yean-Fu Wen and Chih-Lung Chang”Load Balancing Job Assignment for Cluster-Based Cloud Computing”, IEEE ,2014

[2] S.DEVIPRIYA “IMPROVED MAX-MIN HEURISTIC MODEL FOR TASK SCHEDULING IN CLOUD”IEEE 2013

[3] M. Malathi, “Cloud Computing Concepts”, IEEE, 2011

[4] Paul, M., Sanyal, G., "Survey and analysis of optimal scheduling strategies in cloud environment", IEEE, 2012

[5] Jeyarani, R., Ram, R. Vasanth, Nagaveni, N., "Design and Implementation of an Efficient Two-Level Scheduler for Cloud Computing Environment", IEEE, 2010

[6] Huang Qi-yi, Huang Ting-lei, "An optimistic job scheduling strategy based on QoS for Cloud Computing", IEEE, 2010

[7] Meng Xu, Lizhen Cui, Haiyang Wang, Yanbing Bi, "A Multiple QoS Constrained Scheduling Strategy of Multiple Workflows for Cloud Computing", IEEE, 2009

[8] Hao Li, Huixi Li, "A Research of Resource Scheduling Strategy for Cloud Computing Based on Pareto Optimality M×N Production Model",, IEEE, 2011

[9] ” RASA: A New Task Scheduling Algorithm in Grid Environment” Saeed Parsa and Reza Entezari- Maleki World Appl. Sci. J., 7 (Special Issue of Computer & IT): 152-160, 2009.