Review on Max-Min Task scheduling Algorithm for Cloud Computing

Bhavisha Kanani, Bhumi Maniyar
PG students ,Department of Computer Engineering
B.H.College of Engineering and Technology-Rajkot, Gujarat, India

Abstract : Cloud Computing refers as a use of computing ,software, plateform infrastructure as a services via internet.Where customer do not have their own computing infrastructure and pay only for what they use.In such scenario ,task scheduling algorithms can not only meet user's requirements but also get high resource utilization,better performance, quick response time[3].For task allocation cloud provides the virtual machine which is scalable but scheduling them is a major problem.In this paper we study Max-Min task scheduling algorithm[12] and issues related to this algorithm.

Keywords : Cloud Computing, MetaTask Scheduling, RASA Algorithm, Max-min Algorithm, Min-min Algorithm, makespan.

1. INTRODUCTION

Cloud Computing is getting superior day by day. It also make it possible to access applications and associated data from anywhere. Cloud computing can be defined as a collection of computing and communication resources located over distributed datacenters; that is shared by many different users[7]. Some of the drawbacks of the cloud computing is data centre network structure expansibility, energy conservation, replica policies, security and scheduling mechanism[4][5][8]. The primary objective of this paper is optimizing the scheduling polices.
Task scheduling algorithm is a basic requirement To make a number of cloud services for an efficient provider infrastructure. Task scheduling algorithm is responsible for mapping jobs submitted to cloud environment onto available resources in such a way that the total response time ,the makespan is minimized[6].
In cloud computing,There are many tasks require to be executed by the on hand resources to get Minimal total time for completion, Shortest response time, efficient utilization of resources[6][10] etc, Because of these different purpose, we require to design, develop, propose a scheduling algorithm that is used by task scheduler to best appropriate allocation map of tasks on resources.

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 [3], [6], [10], [11], [12].

There are many algorithms used to schedule tasks on their resources, Three well known examples of such algorithms intended to be applied in cloud computing environment are Max-min, Min-min and RASA [1], [3], [11],[12]. Each of these algorithms estimate the completion and execution time of each submitted task on each available resource. RASA is a hybrid algorithm of two other ones. In the RASA,an estimation of the completion time of each task on the available resources is calculated then Max-min and Min-min algorithms are applied alternatively to take advantage of both algorithm and avoids their drawbacks [11].

2. TASK SCHEDULING

Task scheduling is a process for an allocation of one or more time period to one or more resources [6]. In cloud environment,for scheduling scheme, it is difficult to schedule 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.Scheduling is the set of policies to control the order of work to be performed by a computer system.There has been various types of scheduling algorithm existing in distributed computing system, The main advantage of scheduling algorithm is to achieve a high performance computing and the best system throughput[10]. Scheduling manages availability of CPU memory and good scheduling policy gives maximum utilization of resource[6].

Scheduling algorithms can be categorized according to many polices as immediate and batch scheduling, preemptive and non-preemptive scheduling, static and dynamic scheduling[3].

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

In Static Scheduling, Tasks are pre-Schedule, all information are known about available resources and tasks in advanced and a task is assigned once to a resource.

In Dynamic Scheduling, Jobs are available dynamically for scheduling over time by the scheduler. It is more elastic than static scheduling, to be able of determining run time in advance.

In Pre-emptive Scheduling allows each task to be interrupted during execution and a task can be migrated to another resource by leaving its originally allocated resource, available for other tasks. If constraints like priority are considered then this scheduling scheme is more helpful [11].

In Non Pre-emptive Scheduling scheme , resources are not being allowed to be re-allocated until the running and scheduled job finished its execution [11].

3. TERMINOLGY

In the following algorithms denotes number of resources and *s* denotes number of tasks in a meta-task.

Makespan : Makespan is a measure of the throughput of
the heterogeneous computing systems[3].
MET(Minimum Execution Time) : MET assigns each task to the resource that performs it in the least amount of execution time, no matter whether this resource is available or not at that time. This can cause a strict load imbalance across the resources[3].

It takes *O(r)* time to map a given task to an expected resource.[3]

MCT(Minimum Completion Time) : MCT assigns each task to the resource which obtains earliest completion time for that task. This causes some tasks to be assigned to resources that do not have minimum execution time for them. It takes *O(r)* time to map a given task to expected resource, too[3].

In contrast, the Max-min, Min-min and RASA algorithms estimate the execution time and the completion time of each task in meta-tasks then assign the tasks on appropriate resource each based on its decision rule[1][11].

4. EXISTING MAX-MIN ALGORITHM

Min-min scheduling is based on Minimum Completion Time
(MCT) that is used to assign tasks to the resources to have minimum expected completion time. It will work in two Phases, in the first phase, the expected completion time will be calculated for each task in a metatask list. In the second phase, the task with the overall minimum expected completion time from metatask list is select and assigned to the corresponding resource [1][3]. Then this task is removed from metatask list and the process is repeated until all tasks in the metatask list are mapped to the corresponding resources However, the Min- min algorithm is unable to balance the load well as it usually does the scheduling of small tasks initially[12].

The Max-min algorithm is commonly used in distributed environment which begins with a set of unscheduled tasks. Then calculate the expected execution matrix and expected completion time of each task on the available resources. Next, choose the task with overall maximum expected completion time and assign it to the resource with minimum overall execution time. Finally recently scheduled task is removed from the meta-tasks set, update all calculated times, then repeat until meta-tasks set become empty[12].

In the Max-min algorithm, shown in Fig 1, rj represents the ready time of resource Ri to execute a task, while Cij and Eij represent the expected completion time and Execution time respectively. As shown in following fig, task Tm with maximum expected completion time is select to be assigned for related resource Rj that gives minimum execution time[3][12].

```
Step 1: For all submitted tasks in meta-task Ti
Step 2: For all resource Rj
Step 3: Compute Cij = Eij + rj
Step 4: While meta-task is not empty
Step 5: Find the task Tm consumes maximum
          completion time.
Step 6: Assign task Tm to the resource Rj with
          minimum  execution time.
Step 7: Remove the task Tm from meta-tasks
          set
Step 8: Update rj for selected Rj
Step 9:Update Cij for all Ti
```

Fig 1.Max-min algorithm[12]

Max-min algorithm perform better than Min-min algorithm in such situation when the number of short task is more than the longer tasks. For example, if there is only one long task, the Max-min algorithm executes many short tasks concurrently with the long one. In order to avoid the main drawbacks of the Max-min and Min-min, the two schedulers can be executed alternatively to each other for assigning tasks to appropriate Resources, for eliminating each other drawback. Such method, called Resource Awareness Scheduling Algorithm (RASA) which was a new task scheduling algorithm[11].

Max-min algorithm allocates larger task Ti to the resource Rj where large tasks have highest priority rather than smaller tasks [3]. In Max-min we can execute many short tasks concurrently while executing the larger one.
The total makespan is determined by the execution of longer task in this case. But if the metatasks contains tasks with the different completion time then the overall task completion time is not determined by one of the submitted tasks[12].
Proposed algorithm is introduced on the Max-Min algorithm [3][12] in order to reduce the makespan for important jobs and increase the resource utilization.

User-priority was considered in our proposed approach in order to fulfill user's demand quickly for the important job.This algorithm will check first max-min and suppose job is most important than other than set priority of that job so it will execute first.
In the proposed algorithm giving the priority to tasks which have highest and lowest priority. and consider remaining tasks as normal priority tasks.

Makespan is defined as a measure of the throughput of the heterogeneous computing system[3].Where, T represent task, Cij represent completion time of task T with resource j, rj represent resource, Rj represent the ready time of task.

5. PROPOSED ALGORITHM

```
Step 1: Start
Step 2: for all submitted tasks in meta-task Ti
Step 3: for all resource Rj
Step 4: compute Cij = Eij + rj
Step 5: While meta-task is not empty
Step 6: Sort all the tasks in descending order as per Burst Time.
Step 7: Give priority to tasks which have highest and lowest
        priority.
Step 8: Consider remaining tasks as normal priority tasks.
Step 9: Consider three groups according to priority:
                (1)Highest priority group
                (2)Normal priority group
                (3)Lowest priority group
Step 10: First load the tasks of highest priority group and
         arrange task in descending order according to their
         burst time.
Step11: Arrange Vm list in descending order on the basis of
        Resource Cost as
          Resource cost = (RAM of Virtual machine * Cost/memory )
                    +(Size of Virtual machine*Cost/storage)
            (1)  For all tasks in highest priority group Ti,
            (2)  Select resource from resource list sequentially
            (3)  Select task sequentially and schedule on selected resource
            (4)  Remove Task Tk from Meta-tasks set.
            (5)  Update ready time rj for select Rj
                 Update Cij for all Ti
Step 10: Repeat above steps till the each and every task executed
         completely in highest priority group.

Step 11: Repeat above step 9 first for Normal priority group and
         at last for lowest priority group until each and every
         task executed completely.

Step 12: End
```

6. CONCLUSION AND FUTURE WORK

When we use Max-min algorithm always largest task will be assigned to the best available resource (fastest resource) and does not consider the completion time.But, when we schedule tasks using improved max-min task scheduling then largest task is too large compared to other tasks in Meta-task in this case overall makespan is increased because too large task is executed by slowest resource first while other tasks are executed by faster resource. Proposed algorithm is introduced to avoid drawbacks of original Max-Min algorithm in order to reduce the makespan and increase the resource utilization with considering user priority, so that the import job will execute first according to their priority. The job which have higher priority will execute first than other lower priority job so that user's demand can be satisfied more completely.

REFERENCES

[1] Etminani .K, and Naghibzadeh. M, "A Min-min Max-min Selective Algorithm for Grid Task Scheduling," The Third IEEE/IFIP International Conference on Internet, Uzbekistan, 2007.

[2] El-Sayed T. El-kenawy, Ali Ibraheem El-Desoky,etc.al.,"Extended Max-Min Scheduling Using Petri Net and Load Balancing", International Journal of Soft Computing and Engineering (IJSCE),September 2012.

[3] Xiaofang Li, Yingchi Mao, Xianjian Xiao, Yanbin Zhuang," An Improved Max-Min Task-Scheduling Algorithmfor Elastic Cloud" IEEE International Symposium on Computer, Consumer and Control, 2014.

[4] Anu Gupta "Cloud Computing Growing Interest and Related Concerns" IEEE 2nd International Conference on Computer Technology and Development (ICCTD) 2010.

[5] Kresimir Popovic, Zeljko Hocenski" Cloud computing security issues and challenges" MIPRO May 24-28, 2010.

[6] Pham Phuoc Hung, Mui Van Nguyen, Mohammad Aazam, Eui-Nam Huh"Task Scheduling for Optimizing Recovery Time in Cloud Computing" IEEE, 978-1-4799-2903-0/14/$31.00 ©2014.

[7] Peter mell,Timothy Grance "The NIST Definition of Cloud Computing" NIST Special Publication 800-145 September 2011.

[8] Hassan Takabi , James B.D. Joshi" Security And Privacy Challenges In Cloud Computing Environments" Copublished By The IEEE Computer And Reliability Societies, November/December 2010.

[9] Rajender Kumar Trivedi, Rajani Sharma, " Case Study on Environmental Impact of Cloud Computing" IOSR Journal of Computer Engineering (IOSR-JCE), Volume 16, Issue 2, Ver. VI ,Mar-Apr. 2014.

[10] Harpreet Kaur,Maninder Singh" Review of Various Scheduling Techniques in Cloud Computing" International Journal of Networking & Parallel Computing, Volume 1, Issue 2, November, 2012.

[11] Saeed Parsa, Reza Entezari-Maleki," RASA: A New Grid Task Scheduling Algorithm" International Journal of Digital Content Technology and its Applications Volume 3, Number 4, December 2009.

[12] S.Devipriya, C.Ramesh," Improved Max-Min Heuristic Model For Task Scheduling In Cloud" IEEE 2013.

[13] Rajkumar Buyya, Rajiv Ranjan and Rodrigo N. Calheiros," Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities" IEEE, 2009.