Enhanced Heuristic Model for Effective Resource Utilization in Cloud

S. Devipriya
PG Scholar / CSE
Bannari Amman Institute of Technology
Sathyamangalam
India

B. Magesh Kumar
PG Scholar/CSE
Bannari Amman Institute of Technology
Sathyamangalam
India

C. Ramesh
Asst Prof (Sr.Grade)/CSE
Bannari Amman Institute of Technology
Sathyamangalam
India

ABSTRACT
A cloud is a type of distributed system, it consist of, collection of interconnected and virtualized computers. It offers pool of resources like data, software and infrastructure etc, to the user. So the efficient utilization of cloud resources has become a major challenge in cloud computing. Scheduling in cloud is responsible for selection of best suitable resources for executing a task, by considering some static and dynamic parameters like makespan, cost, resource utilization; speed etc.The Min-Min algorithm first executes smaller tasks [1]. So the Min-Min algorithm does not provide better performance when the number of smaller tasks is high, in this case the Max-Min algorithm outperforms Min-Min algorithm in terms of parameters like makespan and load balancing. So in order to overcome the limitations of Min-Min algorithm the improved version of Min-Min algorithm has been proposed. It randomly selects the task for execution based on the values of average completion time and standard deviation of existing tasks. Even though the improved Min-Min algorithm avoids the limitations of Min-Min algorithm; still it takes more time to execute large number of tasks. So the future work is to enhance the performance of improved Min-Min algorithm to execute large number of tasks within in a small time and also to use this algorithm in any one of the network technique to schedule packets in a cloud environment in an effective manner.

Keywords
Cloud computing, Max-Min Algorithm, Min-Min Algorithm, makespan

1. INTRODUCTION
Cloud computing relies on sharing of resources to achieve coherence and economies of scale similar to a utility over a network. Cloud Computing is getting popular every day. Cloud service providers provide services to large scale cloud environment with cost benefits. Also, there are some popular large scaled applications like social networking and internet commerce. These applications can provide benefit in terms of minimizing the costs using cloud computing. Cloud computing is considered as internet based computing service provided by various infrastructure providers based on their need, so that cloud is subject to Quality of Service(QoS), Load Balance (LB) and other factors which have direct effect on user consumption of resources controlled by cloud infrastructure [3].In cloud scheduling process need to achieve several factors. So it needs to use effective algorithm for allocating proper task to the proper resources. Various task scheduling algorithms has been proposed by researchers, most important task scheduling algorithms are Min-Min, Max-Min etc.

2. SCHEDULING PROCESS IN CLOUD
The main advantage of job scheduling algorithm is to achieve a high performance computing and the best system throughput. The available resources should be utilized efficiently without affecting the service parameters of cloud. Scheduling process in cloud can be categorized into three stages they are Resource discovering and filtering, Resource selection, and Task submission. In resource discovery datacenter broker discovers the resources present in the network system and collects status information related to them. During resource selection process target resource is selected based on certain parameters of task and resource. Then during task submission task is submitted to selected resources.

3. EXISTING HEURISTIC MODELS FOR TASK SCHEDULING
Heuristics are applied to find an optimal solution to the scheduling. Heuristic can be both static and dynamic. Static heuristic is applied when the numbers of tasks to be completed are known in prior. Dynamic heuristic can be used when the task arrival is dynamic in nature. This paper focus on heuristic algorithms such as Min-Min, Max-Min and Improved Min-Min. The dynamic heuristic is used in two ways [10]. They are On-line mode and batch mode. In online mode heuristic algorithms tasks are scheduled when they arrive in the system. In batch mode heuristic algorithm tasks are queued and collected into a set when they arrive the system.

3.1 Min-Min Algorithm
Min-Min algorithm is a type of batch mode scheduling algorithm, so all the arriving tasks are grouped in a queue then it is scheduled; group of tasks is called metatask. Min-Min algorithm first executes minimum sized tasks with resource which gives the minimum execution time so it is called as Min-Min algorithm .Min-Min scheduling is based on Minimum Completion Time (MCT); that is used to assign tasks to the resources having minimum expected completion time [14]. Initially a scheduler takes a set of unmapped/ unscheduled tasks and a set of available resources starting by task with minimum MCT to be mapped to next resources; this process is repeated until the unmapped set becomes empty. Briefly, expectation time produces a smaller makespan, which is a measure of throughput of heterogeneous computing systems like computational grids and If more tasks can be obtained, they can be mapped to resources to complete them earliest and executing them faster.

1. for all tasks Ti in meta-task Mv
2. for all resources Rj
3. Cij = Eij + rj
4. do until all tasks in Mv are mapped
5. for each task in Mv find the earliest completion time and the resource that obtains it.
6. find the task Tk with the minimum earliest completion time.
7. assign task Tk to resource that gives the earliest completion time.
8. delete task Tk from Mv .
9. update rj
10. update Cij for all i.
11. end do.

Fig1. Min-Min algorithm

In Fig 2 rj represents the ready time of the resource Rj to execute a task, Cij and Eij represent the expected completion time and execution time of the tasks.

3.2 Max-Min Algorithm

It is also a type of batch mode scheduling algorithm it overcomes the limitations of Min-Min algorithm .Similar to Min-Min, a scheduler schedules tasks by expecting the Execution Time of the tasks and allocation of resources. Instead of selecting the minimum MCT, the maximum MCT is selected, that is why it is named Max-Min [9]. It focuses on giving priority to large tasks over others small. The Max-Min algorithm is typical to the Min-Min algorithm, except for being different in; the word "minimum" would be replaced by "maximum" .Officially Max-Min algorithm does better than Min-Min algorithm in cases when the number of short tasks is more than the longer ones. For example, if there is only one long task, the Max-Min algorithm executes many short tasks concurrently with the long one.

1. for all tasks Ti in meta-task Mv
2. for all resources Rj
3. Cij = Eij + rj
4. do until all tasks in Mv are mapped
5. for each task in Mv find the earliest completion time and the resource that obtains it.
6. find the task Tk with the maximum earliest completion time.
7. assign task Tk to resource that gives the earliest completion time.
8. delete task Tk from Mv .
9. update rj
10. update Cij for all i.
11. end do.

Fig2.Max-Min algorithm

In algorithm 2, the expected time of resource Rj is the time to become ready to execute a task after finishing the execution of all tasks assigned to it which is denoted by rj. Also Eij is the estimated execution time of task Ti on resource Rj whereas Cij is the Expected Completion Time that is the estimated execution time and ready time together.

3.3 Drawbacks

1. Min-Min algorithm attempts to assign the short tasks before the long one.
2. The Min-Min algorithm seems worse in the cases when the number of short tasks is much more then the long ones.
3. Max- Min algorithm can executes short tasks concurrently with the long task.

4. PROPOSED HEURISTIC ALGORITHM FOR TASK SCHEDULING

A typical data centre which consists of commodity machines connected through various high speed links in cloud computing [13]. This environment is used to compute dissimilar and large group of tasks. Basically it is not possible to distinguish the tasks of one user from the other. Hence scheduling problem over here is going to be matching multiple tasks to multi machines. The optimal matching is an optimization problem which possesses NP-complete complexity. In general the heuristic mode is applied as a suboptimal algorithm in order to obtain good solution. In dynamic batch mode heuristic scheduling tasks are queued and collected into a set when they arrive in the system. The scheduling algorithm will start after a fixed period of time and according to some priority factor [10]. This paper focused on most popular existing heuristic approach named Min-Min, in order to overcome the limitations of Min-Min algorithm the improved version of the Min-Min algorithm has been developed.

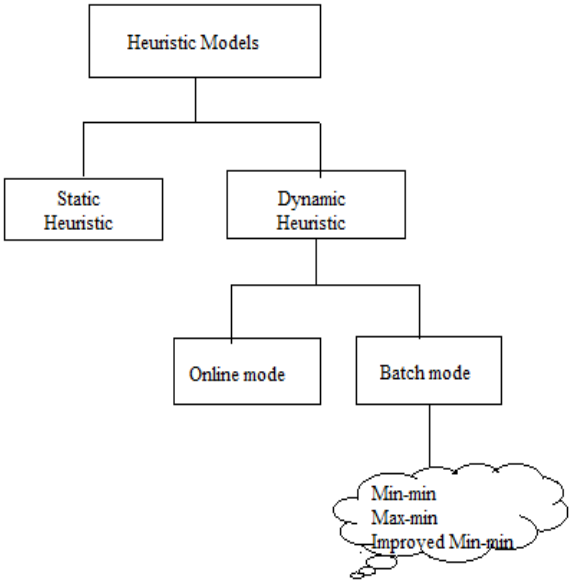

Fig3. Heuristic Models

4.1 Improved Min-Min Algorithm

The Min-Min algorithm does not provide better performance when the number of short tasks is high. So the Max-Min algorithm is used, it can execute short tasks concurrently with the long task on different resources, so it can achieve better performance in terms of load balancing it will automatically decrease the makespan [9]. So in order to avoid the drawbacks of Min-Min algorithm improved Min-Min algorithm is used. In this scheduling algorithm all the tasks should be sorted ascending. It means that tasks with minimum completion time are in front of the queue and tasks with maximum completion time are in rear of the queue. Then this algorithm like the Min-Min algorithm, computes minimum completion time of all tasks on available resources. After that, the resource according to the appropriate condition should be chosen. For selecting a task for scheduling, it first computes the average of completion time and standard deviation of existing tasks.

FLOW CHART

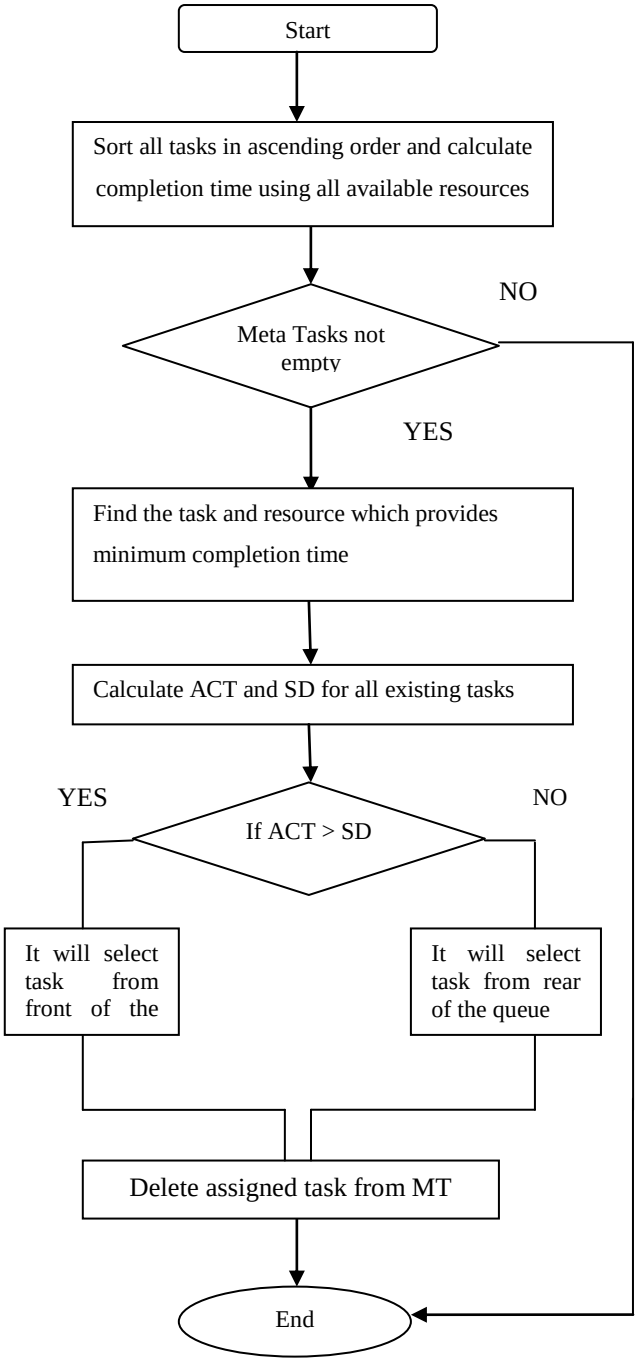

Fig4.Flowchart

The average completion time and standard deviation are calculated using the following formulas

$$ACT = \frac{\sum_{i=1}^{r} CT_{ij}}{r}$$

Where ACT is the average completion time and CT_{ij} is the completion time of task i on resource j, r is the number of resources used.

$$SD = \sqrt{\frac{\sum_{i=1}^{r} (CT_{ij} + ACT)^2}{r}}$$

Where SD it the standard deviation, then ACT is the average completion time, CT_{ij} is the completion time of task i on resource j and r is the number of resources used.

1. Starts with set of unscheduled tasks
2. It first sort all tasks in ascending order
3. Calculate completion time for each task in all resources
4. Find the task and resource which provides minimum completion time
5. Calculate ACT and SD for all existing tasks
6. Compare the values of ACT and SD
7. If ACT > SD, it means the length of all tasks in MT is in a small range, so it will select task from front of the queue.
8. If ACT < SD, it means the length of all tasks in MT is high, so it will select task from rear of the queue.
9. Delete assigned task from MT
10. Repeat these steps until metatask list will become empty

Fig5.Improved Min-Min algorithm

Where
ACT=Average Completion Time
MT=Meta Task
SD=Standard Deviation

The algorithm in fig 5 shows the process involved in improved Min-Min algorithm. It starts with set of unscheduled tasks. Then all the tasks are sorted in ascending order, and then it finds the tasks and the resource which provides minimum completion time, in the next step the values of ACT and SDare calculated in order to schedule the tasks in a random order. The values of ACT and SD are compared, if the value of ACT is greater than SD then it will select task from front of the queue, Otherwise it will select the task from rear of the queue. This process will repeat until all the tasks in Metatask will became empty. Then all the mapped tasks are deleted from the queue.

5. EXPERIMENTAL RESULT

The performance of scheduling algorithms are calculated based on the following parameter

- Makespan or completion time
- Load balancing

Makespan or completion time represent the total time needed to execute the task by the available resource [4]. Load balancing is the factor which is used to perform effective resource utilization. Min-Min algorithm works better if the number of smaller tasks is less and Max-Min algorithm works better if the number of larger tasks is less. To compare and evaluate the proposed algorithm with other algorithms such as Min-Min and Max-Min, a simulation environment known as cloudsim toolkit has been used. Here the number of tasks is given by considering various load conditions like light, medium and heavy. Then the corresponding makespan produced by all the three algorithms are given. Performances of the algorithms are compared based on the values given in table1.The graphical representation for makespan comparison is given in fig6.

Table 1 Experimental Result

Number of tasks	Makespan in ms		
	Max-Min	Min-Min	Improved Min-Min
10	40	36	33
25	55	51	45
50	67	63	61

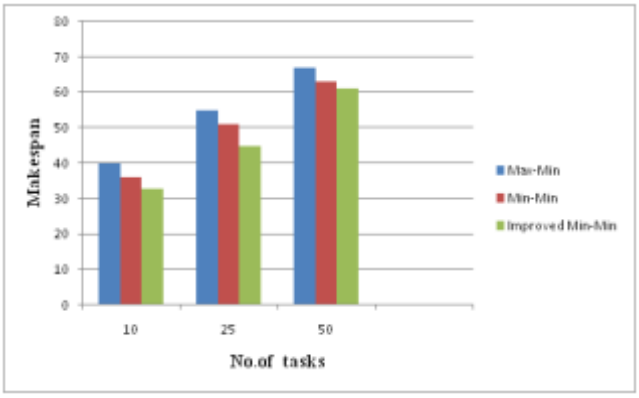

Fig 6.Improved Min-Min algorithm

6. CONCLUSION AND FUTURE WORKS

Now a days the usage of cloud will increase day by day, for optimal use of the capabilities of cloud many effective scheduling algorithms has been proposed [6]. This project mainly focused on well known algorithms such as Min-Min, Max-Min. Min-Min does not work well when the number of smaller tasks are high in this case Max-Min outperforms the Min-Min algorithm [2]. An improved Min-Min algorithm has been developed in order to overcome such limitations of Min-Min algorithm. An improved version of this Min-Min algorithm uses the advantage of Min-Min algorithm and covers their disadvantages. The experimental result obtained by applying these algorithms within cloudsim simulator, shows that the proposed algorithm gives minimum makespan then Min-Min and also it provides load balancing. Future work is to use this algorithm in any one of the network technique to schedule packets in a cloud environment in an effective manner.

7. REFERENCES

[1] He, X., Sun, X.H. & Laszewski, G.V .2003.QoS guided Min-Min heuristic for grid task scheduling. Journal of Computer Science & Technology, vol. 18, pp. 442-451.

[2] Etminani, K., & Naghibzadeh, M. 2007.A Min-Min Max-Min selective algorithm for grid task scheduling. Proceeding of the Third IEEE/IFIP International Conference on Internet, Uzbekista, pp.1-7.

[3] Khoo, B.T., Veeravalli, B., Hung, T. & See, S.W. 2007.A multi-dimensional scheduling scheme in a grid computing environment. Journal of Parallel and Distributed Computing, vol. 67, pp. 659-673.

[4] Mohammad, K. & Analoui, M. 2007.Grid_JQA: A QoS guided scheduling algorithm for grid computing. Proceeding of the Sixth International Symposium on Parallel and Distributed Computing (ISPDC'07), IEEE, pp.34

[5] Munir, E., Li, J. & Shi, S. 2007.QoS sufferage heuristic for independent task scheduling in grid. Information Technology Journal, vol. 6, no. 8, pp. 1166-1170.

[6] Yagoubi, B. & Slimani, Y. 2007.Task load balancing strategy for grid computing. Journal of Computer Science, vol. 3, no. 3, pp. 186-194.

[7] Afzal, A., McGough, A. & Darlington, J. 2008.Capacity planning and scheduling in grid computing environment. Journal of Future Generation Computer Systems, vol. 24, pp. 404-414.

[8] Elmroth, E & Tordsson, J 2008.Grid resource brokering algorithms enabling advance reservations and resource selection based on performance predictions. Journal of Future Generation Computer Systems, vol. 24, pp. 585-593.

[9] Parsa, S. & Entezari-Maleki, R. 2009.RASA: A new task scheduling algorithm in grid environment. In World Applied Science Journal 7, pp.152-160.

[10] FatosXhafa, Ajith Abraham 2010.Computational models and heuristics methods for grid scheduling problems. Future Generation Computer systems, vol. 26, pp. 608-621.

[11] Kokilavani, T. and Dr.George Amalarethinam, D., I., 2011.Load Balanced Min-Min Algorithm for Static Meta-Task Scheduling in Grid Computing.International Journal of Computer Applications, vol. 20, No. 2, pp. 43-49.

[12] Sahu R., Chaturvedi A., K., 2011Many-Objective Comparison of Twelve Grid Scheduling Heuristics.International Journal of Computer Applications ", vol13, pp. 9-17.

[13] Madhubala, R., 2012.An Illustrative study on Cloud Computing.International Journal of Soft Computing and Engineering", vol. 1, issue 6, pp. 286-290.

[14] Anousha, S. & Ahmadi, M. 2013.A batch mode Min-Min scheduling algorithm in grid computing.Springer, pp.103-113.

[15] Pional, S. 2013.A survey of scheduling algorithms in cloud computing environment. International Journal of Research in Engineering and Technology, vol.2, pp.131-135.