International Journal of Science, Environment and Technology, Vol. 3, No 3, 2014, 997 – 1003

ISSN 2278-3687 (O)

A REVIEW ON DYNAMIC FAIR PRIORITY TASK SCHEDULING ALGORITHM IN CLOUD COMPUTING

Deepika Saxena, Scholar[1]
Dr. R.K. Chauhan[2]
Department of Computer Science and Applications
Kurukshetra University, Kurukshetra (Haryana), India
E-mails: 13deepikasaxena@gmail.com[1] / rkchauhandcsakuk@gmail.com[2]

Abstract: Cloud computing, the term in present scenario, covers almost entire internet-based activity. Today, almost everything is deployed, run and managed on the cloud environment. The growing demand for cloud based activities and its day by day increasing count, of number of users and providers, has made it the cynosure among the computing environments. This emerging interest of users and providers has made optimized scheduling of tasks, a major challenge in this arena. For the substantial growth of cloud computing, there is an emerging need of an efficient, optimized and dynamic task scheduling strategy. In this paper we are providing review on different task scheduling algorithms and a review on our proposed task scheduling algorithm named "Dynamic fair priority task scheduling algorithm in cloud computing".

I. INTRODUCTION

Scheduling in a cloud environment is one of the most challenging issues. Scheduling refers to the methodology followed for assigning tasks to machines. Almost all of the desirable qualities like QoS, cost-effectiveness, time-constraint, profit etc. depend on the scheduling strategy to be followed. A scheduling strategy should take care of all of these things. But no such scheduling strategy exists which is concerned with both the user's interest as well as provider's interest. Traditional scheduling strategies like FCFS, SJF, Round-Robin, Grouping etc. are also not good enough to do justice with both user's interest as well as provider's. In this paper, a review on scheduling strategy is devised as a solution to this problem. The strategy is named as, "Dynamic fair priority task scheduling algorithm in cloud computing".

This review paper covers the introduction in first section, second section discusses about the related work so far in this area and third section discusses the proposed strategy. And finally last section discusses about conclusion and references.

II. RELATED WORK

Scheduling of tasks and resources has always been the prime concern in the performance of any scheduling strategy. Traditional scheduling strategies used for task scheduling are FCFS,

*Received Apr 23, 2014 * Published June 2, 2014 * www.ijset.net*

SJF, greedy method, prioritization etc. With time, several strategies have been proposed for doing the similar job i.e., scheduling. Each of the strategy proposed so far, has its own pros and cons. scheduling of tasks in a dynamic environment is basically a NP-complete problem. Types of grouping based scheduling was introduced by [9]. Grouping technique reduces the communication overhead of assigning tasks to virtual machines. Dynamic Job Grouping-based scheduling strategy [10], that maximizes the utilization of Grid resource processing capabilities, and reduces the overhead time and cost taken to execute the jobs on the Grid. A scheduling algorithm [11] can minimize average turnaround time in heterogeneous resource environment. This algorithm was based on greedy method which is used for totally dynamic grid environment where the jobs arrive at different time interval. The turnaround time of each job was minimized individually to minimize the average turnaround time of all submitted jobs in a time slot. Dynamic Optimization Algorithm [13] for task scheduling, which groups the tasks based on their type, arranges them in priority queue and schedules tasks. Algorithm attempts to minimize the execution time as well as cost. There is an efficient scheduling algorithm which addresses these major challenges of task scheduling in cloud. The incoming tasks are selected by Data Center Broker on the basis of user's task requirement like minimum execution time or cost and then it is prioritized. So the algorithm is named as "TPD Scheduling Algorithm", where T stands for Task Selection, P Stands for Priority (in terms of cost) and D stands for Deadline [14]. Double Level Priority based algorithm [15] which groups the task on the basis of data and requested resources and prioritize the tasks. Resource selection is done on the basis of its cost and turnaround time both using greedy approach. Task selection on the basis of a priority formula, in this way, it attempts to perform better over sequential approach in terms of cost and execution time.

All of these existing strategies for scheduling are better than the traditional scheduling strategies in one way or the in other. Some are delivering better time bound results; others are concerned with reducing the cost incurred or with the efficiency and several such factors. Although, work is already done towards minimizing the execution time and cost tasks but there are some flaws like few only incorporate grouping technique and then using FCFS which is only reducing the communication overhead, but no work is done to improve execution time and cost. Others which are considering these factors are mostly following grouping followed by prioritization [13], but simple priority suffers from the problem of starvation (long waiting queues). Considering all such issues we can design an algorithm which will attempt to minimize all such flaws and provides better results than existing strategies.

III. PROPOSED STRATEGY

The proposed algorithm will work in four steps. First of all it groups the task into two groups namely, deadline based and cost based to reduce the communication overhead. In second step, the deadline based tasks are arranged in a queue in ascending order of deadline and cost based tasks are arranged in another queue in descending order of task length. In third step, tasks from these queues are mapped into three priority queues namely, HIGH, MID and LOW. At a time three tasks are executed from HIGH list, two from MID list and one from LOW list. Since tasks are executing from all of the three lists, so the basic problem of prioritization i.e., starvation is solved here. In the final step, for deadline based tasks, the turnaround time of all the available virtual machines is calculated, and the task is assigned to virtual machine having lowest turnaround time. Steps are:

(1) Make groups of tasks based on constraints (deadline or cost-based) classification.

(2) For deadline based task group, as shown in fig 1, as below:

(i) Arrange deadline based task in 3 priority queues –high, mid, low.

(ii) Take one task at a time from each priority queue, using weighted fair queue.

(iii) For each VM compute minimum turnaround time as

Turnaround time = waiting time + (task length/processing power)

And update its waiting time accordingly.

(iv) For each task, select VM with minimum turnaround time, so that computation time of each task is reduced individually.

(3) For cost based task group, as shown in fig 2, as below:

(i) Arrange task in high, mid, low priority queues according to descending order of task length.

(ii) Apply weighted fair queue, and take one job at a time.

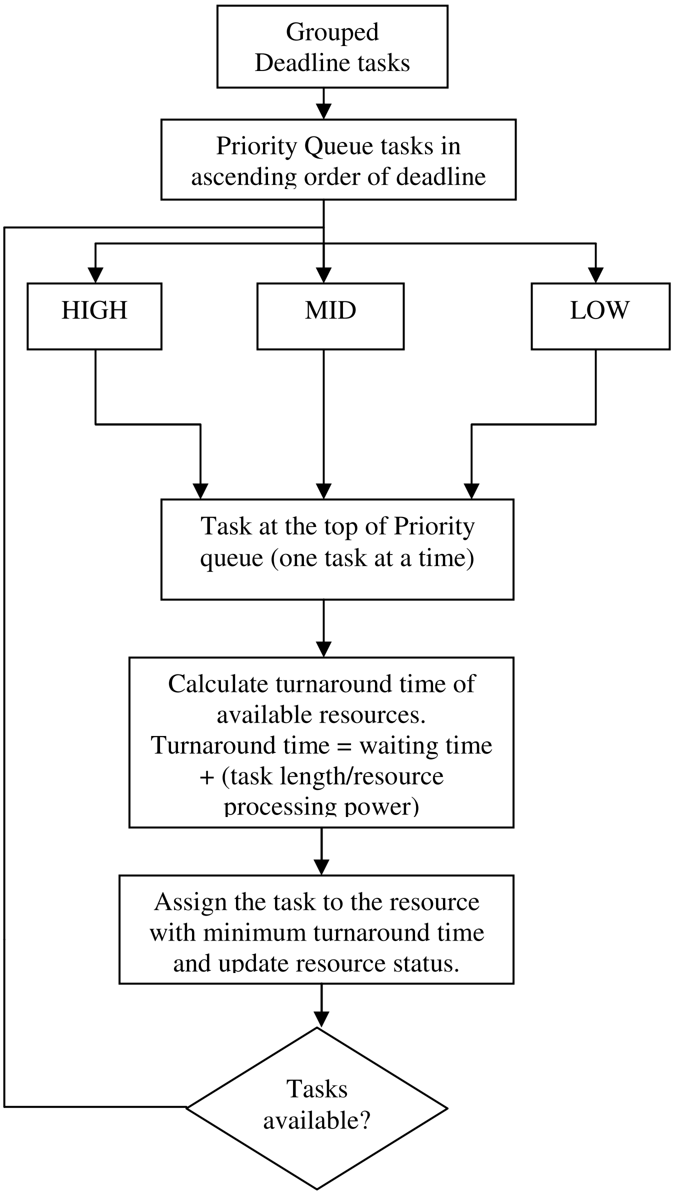
Grouped Deadline tasks
Priority Queue tasks in ascending order of deadline
HIGH
MID
LOW
Task at the top of Priority queue (one task at a time)
Calculate turnaround time of available resources. Turnaround time = waiting time + (task length/resource processing power)
Assign the task to the resource with minimum turnaround time and update resource status.
Tasks available?

Fig.1. Scheduling of Deadline based tasks

(iii) Compute cost of each VM using formula:

Cost = resource cost * (task length/processing power) and sort VM in descending order.

(iv) Assign task to minimum cost VM and task length<= MIPS of VM (so optimization is done at cost level)

In this proposed strategy, we can utilize the advantageous part of all the available strategies as clear from table I and can apply newly designed weighted fair priority in place of simple priority, so that none of the task priority queue suffers from the problem of starvation.

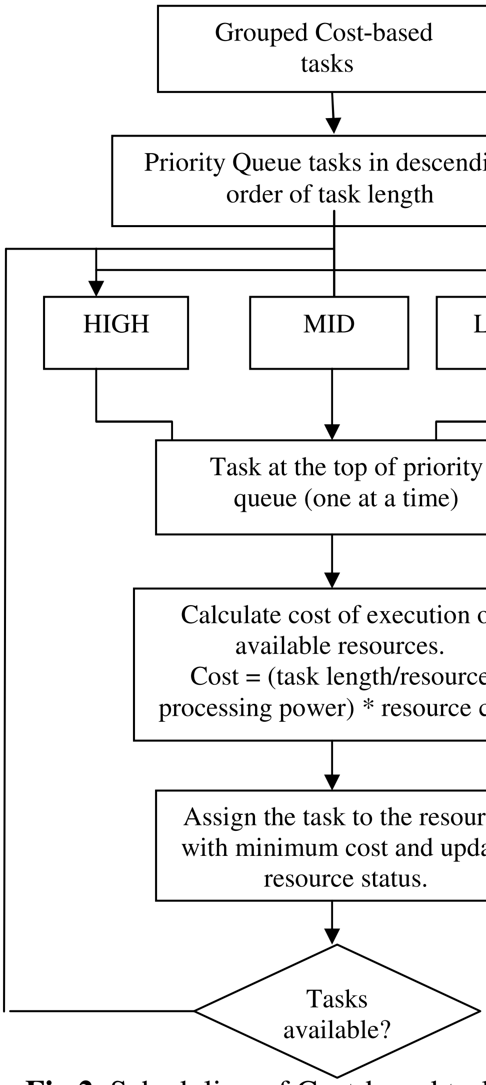
Grouped Cost-based tasks
Priority Queue tasks in descending order of task length
HIGH
MID
LOW
Task at the top of priority queue (one at a time)
Calculate cost of execution of available resources.
Cost = (task length/resource processing power) * resource cost
Assign the task to the resource with minimum cost and update resource status.
Tasks available?

Fig.2. Scheduling of Cost-based tasks

Module	Advantages
Task Grouping	• Reduced communication cost • Easy scheduling and effective resource utilization
Weighted Fair Prioritization	• Task execution in order of High, Mid and Low priority according to assigned weights. • Help to achieve better QoS
Greedy Approach	• Less waiting time and cost

Table I. Scheduling Modules with Advantages

IV. CONCLUSION

Scheduling of task in cloud environment is a highly researched and challenging issue in cloud computing. To meet thousand of service requests while making best possible use of available resources and simultaneously satisfying both user as well as service provider, is the challenge for task scheduler. Traditional methods of scheduling lead to overpricing and slow processing for bulk of tasks. Some task scheduling algorithms are cost based, some are deadline based and many algorithms make use of priority based scheduling. But they suffer from long waiting priority queues. Our proposed algorithm definitely meet all the challenges, along with constraint based optimization scheduling. We have also, introduced fair-priority scheduling concept i.e. combination of priority with round-robin scheduling scheme.

Weighted fair queue brings fairness at priority level and increases utilization of resources at system level and therefore, will provide much more efficient results than that can be provided by any other existing task scheduling algorithm.

REFERENCES

[1] J. Geelan, "Twenty-one experts define cloud computing," *Cloud Computing Journal,* vol. 2009, pp. 1-5, 2009.

[2] R. Mikkilineni and V. Sarathy. (2009). *Cloud Computing and the Lessons from the Past* in 18th IEEE International Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises, WETICE'09, 2009.

[3] S. Ma, "A Review on Cloud Computing Development," *Journal of Networks,* vol. 7, no.2, pp. 305-310, 2012.

[4] P. Mell and T. Grance, "The NIST definition of cloud computing," *National Institute of Standards and Technology,* vol. 53, no.6, 2009

[5] I. Foster, Z. Yong, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," 2008, pp. 1-10.

[6] *Cloud Computing* [Online]: http://en.wikipedia.org/wiki/Cloud_computing

[7] A. Fox and R. Griffith, "Above the clouds: A Berkeley view of cloud computing," *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS,* vol. 28, 2009.

[8] Q. Cao*, et al.*, "An optimized algorithm for task scheduling based on activity based costing in cloud computing," 2009, pp. 1-3.

[9] P.J. Wild*, et al.*, "Understanding task grouping strategies," *PEOPLE AND COMPUTERS,* pp. 3-20, 2004.

[10] N. Muthuvelu*, et al.*, "A dynamic job grouping-based scheduling for deploying applications with fine-grained tasks on global grids," 2005, pp. 41-48.

[11] S. Singh and K. Kant, "Greedy grid scheduling algorithm in dynamic job submission environment," in *Emerging Trends in Electrical and Computer Technology (ICETECT), 2011 International Conference on*, 2011, pp. 933-936.

[12] R.N. Calheiros, *et al.*, "CloudSim: A novel framework for modeling and simulation of cloud computing infrastructures and services," *Arxiv preprint arXiv: 0903.2525,* 2009.

[13] Monika Choudhary, *et al.,* "A Dynamic Optimization Task Scheduling Algorithm in Clou*d Environment"* International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 3, May-Jun 2012, pp.2564-2568 2564

[14] Dr. V. Vaithiyanathan *et al.,* "An Efficient TPD Scheduling Algorithm for Cloud Environment", International Journal of Engineering and Technology (IJET), ISSN: 0975-4024, Vol 5 No 3 Jun-Jul 2013.

[15] S.V. Parikh *et al.,* "Double Level Priority Based Task Scheduling with Energy Awareness in Cloud Computing", Journal of Information, Knowledge and Research in Information Technology, ISSN: 0975 – 6698| NOV 12 TO OCT 13 | VOLUME – 02, ISSUE – 02.