Monika Choudhary, Sateesh Kumar Peddoju / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 3, May-Jun 2012, pp.2564-2568

# A Dynamic Optimization Algorithm for Task Scheduling in Cloud Environment

**Monika Choudhary**
Department of Electronics and Computer Engineering
IIT Roorkee

**Sateesh Kumar Peddoju**
Department of Electronics and Computer Engineering
IIT Roorkee

## ABSTRACT
Cloud computing has emerged as a popular computing model to support on demand services. It is a style of computing where massively scalable resources are delivered as a service to external customers using Internet technologies. Scheduling in cloud is responsible for selection of best suitable resources for task execution, by taking some static and dynamic parameters and restrictions of tasks' into consideration. The users' perspective of efficient scheduling may be based on parameters like task completion time or task execution cost etc. Service providers like to ensure that resources are utilized efficiently and to their best capacity so that resource potential is not left unused. This paper proposes a scheduling algorithm which addresses these major challenges of task scheduling in cloud. The incoming tasks are grouped on the basis of task requirement like minimum execution time or minimum cost and prioritized. Resource selection is done on the basis of task constraints using a greedy approach. The proposed model is implemented and tested on simulation toolkit. Results validate the correctness of the framework and show a significant improvement over sequential scheduling.

**Keywords**
Cloud Scheduling, Optimal Scheduling, Dynamic task execution

## 1. INTRODUCTION
Cloud computing is a very current topic and the term has gained a lot of attention in recent times. It can be defined as on demand pay-as-per-use model in which shared resources, information, software and other devices are provided according to the clients' requirement when needed [1]. Human dependency on cloud is evident from the fact that today's most popular social networking, email, document sharing and online gaming sites are hosted on cloud. Google, Microsoft, IBM, Amazon, Yahoo and Apple among others are very active in this field.

Scheduling theory for cloud computing is receiving growing attention with increase in cloud popularity. In general, scheduling is the process of mapping tasks to available resources on the basis of tasks' characteristics and requirements. It is an important aspect in efficient working of cloud as various task parameters need to be taken into account for appropriate scheduling. The available resources should be utilized efficiently without affecting the service parameters of cloud.

Scheduling process in cloud can be generalized into three stages nam**e**ly–

- Resource discovering and filtering – Datacenter Broker discovers the resources present in the network system and collects status information related to them.
- Resource selection – Target resource is selected based on certain parameters of task and resource. This is deciding stage.
- Task submission -Task is submitted to resource selected.

The simplified scheduling steps mentioned above are shown in Figure 1

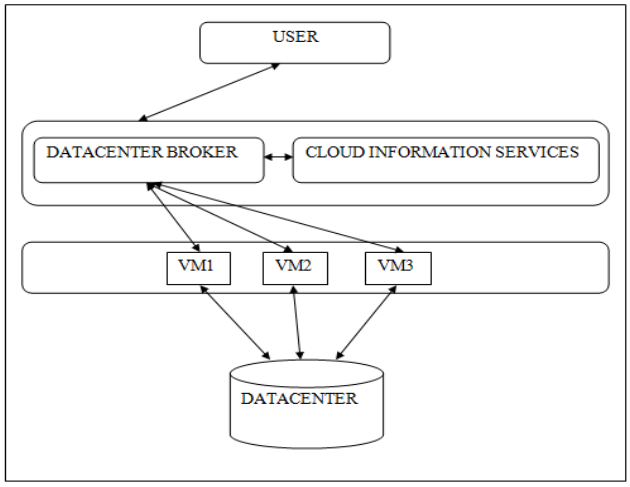


**Figure 1. Scheduling in Cloud**

The rest of this paper is organized as follows: Section 2 briefly discusses related work followed by proposed framework in Section 3. Next, Section 4 presents the proposed scheduling algorithm and its strategy. In Section 5 the experimental details and results of experiments are presented with comparison with some existing algorithms. Finally, Section 6 concludes the paper and proposes future improvements.

## 2. Related Work

Target resources in a cloud environment can be selected in various ways. The selection of resources can be either random, round robin, greedy (resource processing power and waiting time based) or by any other means. The selection of jobs to be scheduled can be based on FCFS, SJF, priority based, coarse grained task grouping etc. Scheduling algorithm selects job to be executed and the corresponding resource where the job will be executed. As each selection strategy is having certain flaws work could be done in this direction to extract the advantageous points of these algorithms and come up with a better solution that tries to minimize the drawbacks of resultant algorithm.

The existing algorithms are beneficial either to user or to cloud service providers but none of them takes care of both. Each have their own advantages and disadvantages. Like greedy and priority based scheduling are beneficial to user and grouping based scheduling is concerned with better utilization of available resources. But the priority based scheduling may lead to long waiting time for low prioty tasks. Greedy scheduling from users point of view lead to wastage of resources whereas greedy scheduling from service providers point of view may lead to disappointment for user on QoS parameters. Similarly task grouping may have the disadvantage of considerable task completion time due to formation of groups. Thus we see that some scheduling strategies are biased to users while others to service providers. There is an emerging requirement to balance this biasing to form an optimized scheduling solution.

New scheduling strategy need to be proposed to overcome the problem posed by network properties and user requirements. The new strategies may use some of the conventional scheduling concepts to merge them with some network and requirement aware strategies to provide solution for better and more efficient task scheduling.

## 3. Proposed Framework

**Task Grouping:** Grouping means collection of components on the basis of certain behavior or attribute. By task grouping in cloud it is meant that tasks of similar type can be grouped together and then scheduled collectively [2]. We can say that it is a behavior that supports the creation of 'sets of tasks' by some form of commonality. In the proposed framework tasks are grouped on the basis of constraint which can be deadline or minimum cost. Once the tasks are grouped, they can be judged for their priority and scheduled accordingly. Grouping, if employed to combine several tasks, reduces the cost-communication ratio.

**Prioritization**: Priority determines the importance of the element with which it is associated. In terms of task scheduling, it determines the order of task scheduling based on the parameters undertaken for its computation [3]. In the present framework, the deadline based tasks are prioritized on the basis of task deadline. The tasks with shorter deadline need to be executed first. So they are given more priority in scheduling sequence. The task list is rearranged with tasks arranged in ascending order of deadline in order to execute the task with minimum time constraint first. The cost based tasks are prioritized on the basis of task profit in descending order. This is appreciable as tasks with higher profit can be executed on minimum cost based machine to give maximum profit.

**Greedy Allocation** : Greedy algorithm is suitable for dynamic heterogeneous resource environment connected to the scheduler through homogeneous communication environment [4]. Greedy approach is one of the approach used to solve the job scheduling problem.

According to the greedy approach -

"*A greedy algorithm always makes the choice that looks best at that moment. That is, it makes a locally optimal choice in the hope that this choice will lead to a globally optimal solution*" [5].

**Deadline Constrained Based** - To improve the completion time of tasks greedy algorithm is used with aim of minimizing the turnaround task of individual tasks, resulting in an overall improvement of completion time.

*Turnaround Time = Resource Waiting Time + Task Length /*

*Proc. Power of Resource*

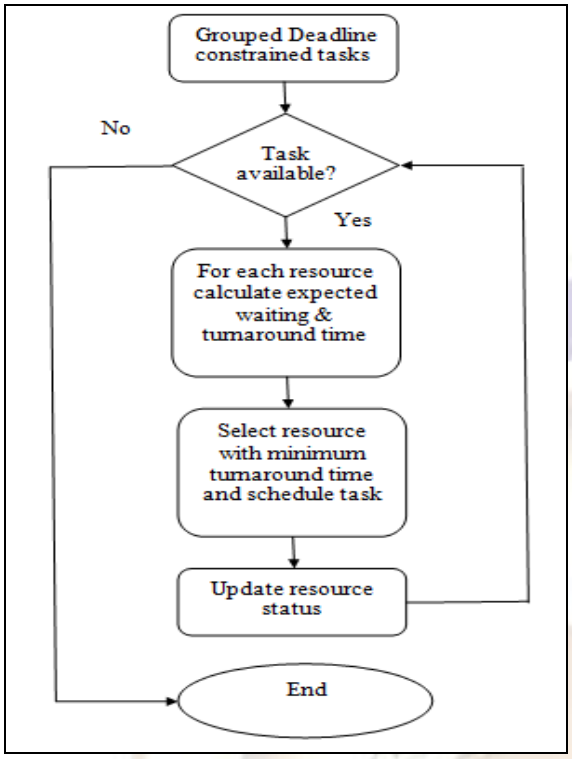


**Figure 3.1 Scheduling of Deadline Constrained Tasks**

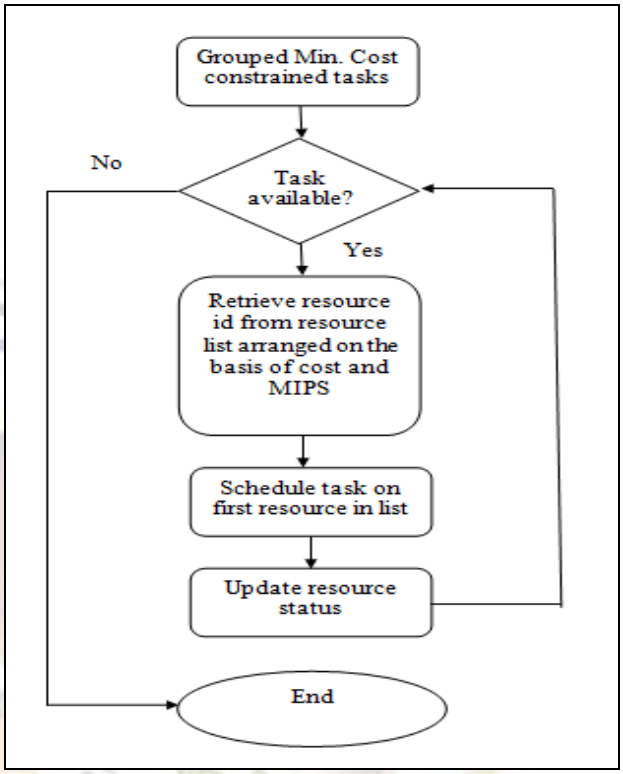


**Figure 3.2 Scheduling of Cost Based Tasks**

After calculating the turnaround time for each resource, the resource with minimum turnaround time is selected and task is executed there. The scheduler locates the best suited resource that minimizes the turnaround time. The turnaround time is calculated on the basis of expected completion time of a job. Once the scheduler submits a task to a machine, the resource will remain for some time in processing of that job. The resource status is updated to find out when the resource will be available to process a new job.

**Minimum Cost Based** - The resource with minimum cost is selected and tasks are scheduled on it until its capacity is supported. After scheduling each task the resource status is updated accordingly. Thus the selection of task and target resource is sequential once they are prioritized according to user needs.

*Cost of Task = (Task length / Proc Power of Resource) **

*Resource Cost*

## 4. Proposed Algorithm

An optimum scheduling algorithm is proposed and implemented in this section. The proposed algorithm works as follows

1. Incoming tasks to the broker are grouped on the basis of their type– deadline constrained or low cost requirement.
2. After initial grouping they are prioritized according to deadline or profit. This is required because the tasks with shorter deadline need to be scheduled first and similarly the tasks resulting in more profit should be scheduled on lost cost machines. Thus, the prioritizing parameter is different based on the nature or type of task.
3. a. For each prioritized task in deadline constrained group –
   i) Turnaround time at each resource is calculated taking following parameters into account.
      - Waiting time
      - Task length
      - Processing Power of virtual machine

   ii) The virtual machine with minimum turnaround time that is capable to execute the task is selected and task is scheduled for execution on that machine.

iii) Waiting time and resource capacity of selected machine are updated accordingly.

b. For cost based group

i) Virtual Machine are selected on the basis of processing power of machine and its cost

ii) For each virtual machine cloudlets from the group are scheduled till the resource capacity is permitted.

iii) Resource capacity and waiting time are updated accordingly.

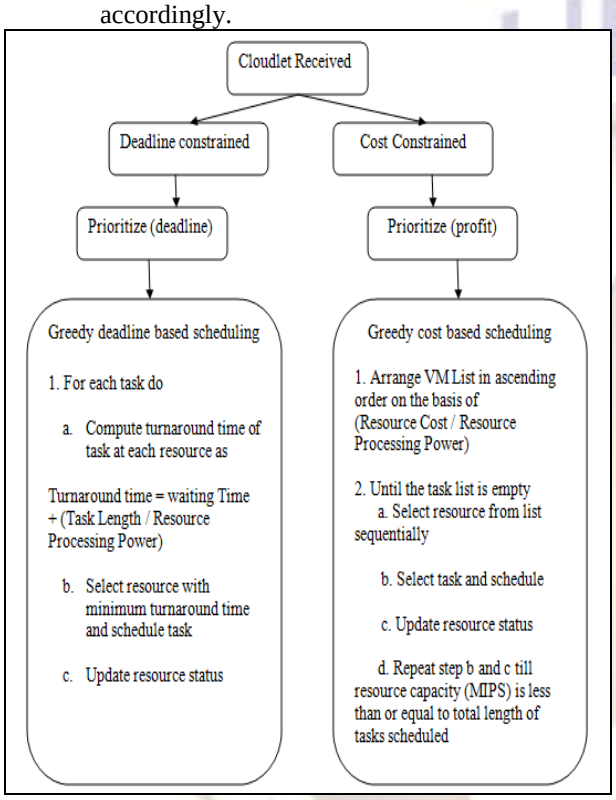


**Figure 4.1 Proposed Algorithm**

## 5. Simulation Results

The CloudSim toolkit is used to simulate heterogeneous resource environment and the communication environment [6,7]. CloudSim(2.1.1) simulator is used to verify the correctness of proposed algorithm. The experiments are performed with Sequential assignment which is default in CloudSim and the proposed algorithm. The jobs arrival is Uniformly Randomly Distributed to get generalized scenario.

The configuration of datacenter created is as shown below - Number of processing elements – 1 Number of hosts – 2

**Table 1 Configuration of Hosts**

| | |
|---|---|
| RAM(MB) | 10240 |
| Processing Power(MIPS) | 110000 |
| VM Scheduling | Time Shared |

The configuration of Virtual Machines used in this experiment is as shown in Table 3.

**Table 2 Configuration of VMs**

| Virtual Machines | VM 1 | VM2 |
|---|---|---|
| RAM(MB) | 5024 | 5024 |
| Processing Power(MIPS) | 22000 | 11000 |
| Processing Element(CPU) | 1 | 1 |

**Performance with cost:** The tasks execution using the proposed algorithm results in a significant improvement in cost over the sequential allotment as shown in Table 3.

**Table 3 Comparison of Execution Cost**

| No. Of Cloudlets | Proposed Algorithm | Sequential Assignment |
|---|---|---|
| 25 | 565.91 | 735.68 |
| 50 | 1131.82 | 1471.36 |
| 75 | 1697.73 | 2207.05 |
| 100 | 2263.6 | 2942.73 |

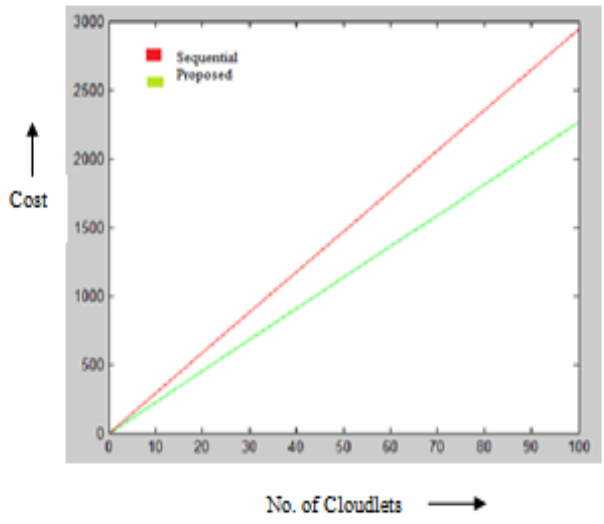

**Figure 5.1 Analysis of Execution Cost**

**Performance with time:** It is evident from the results that proposed algorithm gives better completion time of job in comparison to the sequential approach.

**Table 4 Comparison of Task Completion Time**

| Cloudlets | Proposed Algo | Sequential Algo |
|---|---|---|
| 25 | 565.91 | 735.68 |
| 50 | 1131.82 | 1471.36 |
| 75 | 1697.73 | 2207.05 |
| 100 | 2263.6 | 2942.73 |
| 125 | 910.04 | 997.99 |
| 150 | 1298.50 | 1439.75 |

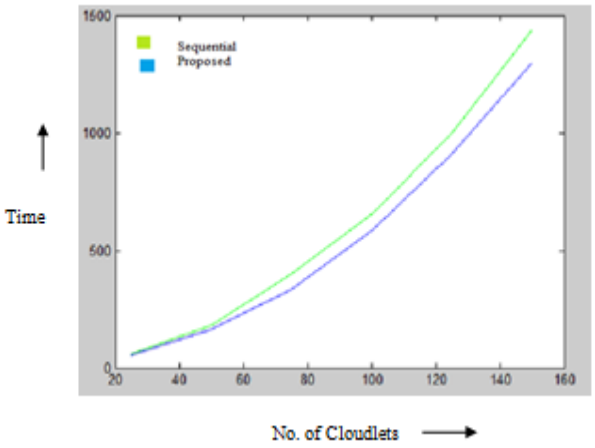


**Figure 5.2 Analysis of Task Completion Time**

## 6. CONCLUSION AND FUTURE WORK

It is observed that the proposed algorithm improves cost and completion time of tasks as compared to Sequential Assignment. The turnaround time and cost of each job is minimized individually to minimize the average turnaround time and cost of all submitted tasks in a time slot respectively. The results improve with the increase in task count.

The proposed algorithm can be further improved by considering following suggestions -

- The future work may group the cost based tasks before resource allocation according to resource capacity to reduce the communication overhead.
- Other factors like type of task, task length could be taken into account for proper scheduling of tasks.

## REFERENCES

[1] J. Geelan, "Twenty-one experts define cloud computing," *Cloud Computing Journal*, vol.4, pp. 1-5, 2009.

[2] P. J. Wild, P. Johnson and H. Johnson, "Understanding task grouping strategies," *PEOPLE AND COMPUTERS*, pp. 3-20, 2004.

[3] Q. Cao, B. Wei and W. M. Gong, "An optimized algorithm for task scheduling based on activity based costing in cloud computing," In International Conference on eSciences 2009, pp. 1-3.

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

[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein *Introduction to algorithms*: The MIT press, 2001, pp 16

[6] R. N. Calheiros, R. Ranjan, C. A. F. De Rose, and R. Buyya, "Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services," *Arxiv preprint arXiv:0903.2525*, 2009.

[7] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose and R. Buyya "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23-50, 2011.