An Efficient TPD Scheduling Algorithm for Cloud Environment

Dr.V.Vaithiyanathan, R.Arvindh Kumar, S.Vignesh, B.Thamotharan, B.Karthikeyan
School Of Computing, SASTRA University,
Thanjavur, TamilNadu, India
vvn@it.sastra.edu ,arvindh0711@gmail.com, vignesh92s@gmail.com, thamotharan@ict.sastra.edu, karthikeyan@it.sastra.edu

ABSTRACT

Cloud computing is a prominent way to support demand on services. It is a mode of computing where scalable resources are delivered as a service to customers over Internet. Scheduling in cloud is responsible for selection of best suitable resources for task execution, by taking some parameters and restrictions of tasks into consideration. From the users view, efficient scheduling may provide factors like fast service, minimum task execution cost etc. On the other hand Service providers should gain factors like to maximum profit, utilize their service efficiently and importantly regular customers. This paper proposes an efficient scheduling algorithm which addresses these major challenges of task scheduling in cloud. The incoming tasks/users can select their method on the basis of task requirement like minimum execution time or cost and then it is prioritized. So the algorithm is named as “TPD Scheduling Algorithm”, Here T stands for Task Selection, P Stands for Priority(in terms of cost) and D stands for Deadline. The proposed model is implemented and tested on simulation toolkit. Results validate the correctness of the framework and show a significant improvement over other scheduling methods.

Keywords: Cloud Computing, Efficient Scheduling, Deadline, Priority, Cost Based

I. Introduction

Cloud Computing is a class of network based computing that takes place over the Internet, basically an enhanced step of Utility Computing. This is a collection of integrated hardware, software and Internet infrastructure. Hardware, software and networking services to clients can be provided by using the Internet. A simple Graphical/Application programming interface can be used to hide the complexity and background infrastructure from the users and the application in these platforms. In addition, the platform provides On demand- Ubiquitous services[1,2]. Scheduling plays a vital role in cloud for selection of best suitable resources for task execution to prevent the file system corruption and to reduce the probability of data loss. The main parameters considered for scheduling are task grouping, priority based, minimum execution time/cost and deadline constraint.

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

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

The simplified scheduling steps mentioned above are shown in Figure 1

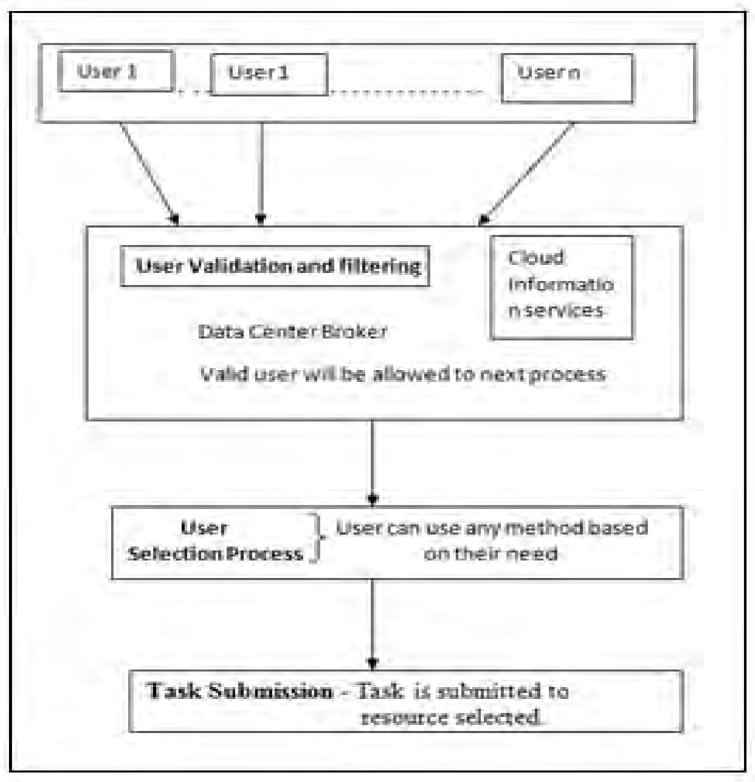

User 1
User1
User n
User Validation and filtering
Cloud Informatio n services
Data Center Broker
Valid user will be allowed to next process
User Selection Process
User can use any method based on their need
Task Submission - Task is submitted to resource selected

Figure 1. Scheduling Process in Cloud

The rest of this paper is structured 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. Finally, Section 6 concludes the paper and proposes future improvements.

II Related Work

The selection of jobs to be scheduled can be based on FCFS, SJF, priority based,etc[6]. 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 priority based scheduling are beneficial to user and scheduling is concerned with better utilization of available resources. But the priority based scheduling may lead to long waiting time for low priority tasks. Similarly task grouping may have the disadvantage of considerable task completion time due to formation of groups[4].Thus 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.

Proposed Framework:

User selection: This means Selection of components on the basis of certain behaviour or attribute. By user selection in cloud it is meant that tasks of similar type can be selected together and then scheduled collectively [3]. It is a behaviour that supports the creation of ‘sets of tasks’ by some form of commonality. In the proposed framework user selects the method on the basis of constraint which can be deadline and cost. Once the method selected, they can be judged for their priority and scheduled accordingly.

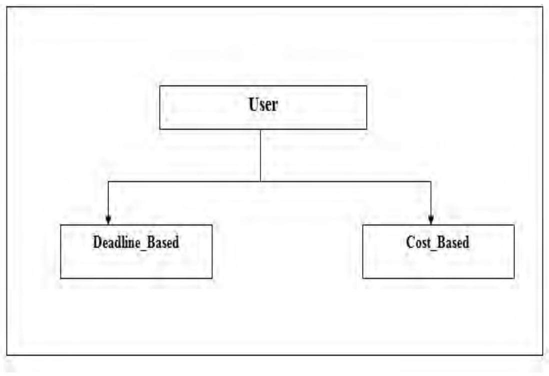


Figure-2 User Selection

Deadline Based Scheduling: Once the users selects deadline based scheduling, they can use the apps only for certain time,extend of time will not be allowed. If the time over, it will automatically gets logout. Coming to the cost side its common for all users because all of them having same access time.

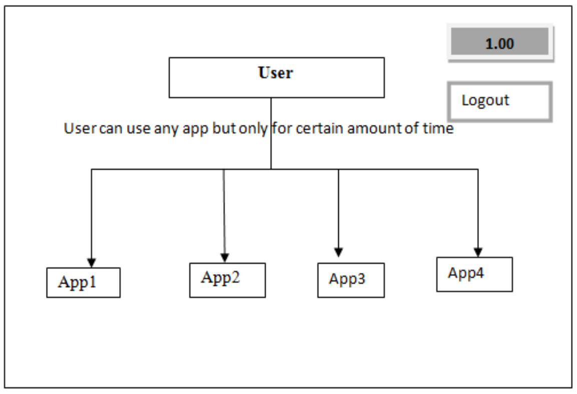


Figure-3 Deadline Based

Cost Based Scheduling : If the users selects the cost based scheduling, they can use their app for as per their wish i.e there is no time costraint. Here coming to the cost side, they should pay for how much they used.Once they finish their work they can come out of app by logout.

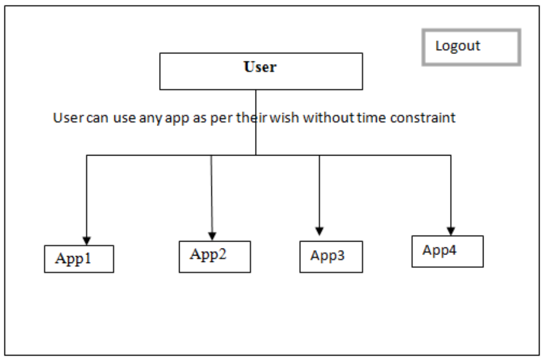

Logout
User
User can use any app as per their wish without time constraint
App1
App2
App3
App4

Figure-4 Cost Based Scheduling

Proposed Algorithm:

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

1). User send all request to cloud,

2).A valid incoming users are allowed to select one method on the basis of their type- deadline based or cost based

3). Once the user selects any one method, do the following,

i) For Dead line based,

For each user request do

- (i)Compute Start time of each user(first come first serve)

  (ii) Select user with minimum start time and schedule the task.

  (iii) Update the application status.
- (i)If some users having same start time do the following , Check the count(i.e., check whether user already used the cloud)

  (ii) Select user with maximum count and schedule the task.

  (iii) Update the application status.
- (i)if some users having same count, use cost based count(Check he used cost based method)

  (ii) Select user with maximum cost based count and schedules the task.

  (iii) Update the application status.
- If cost based count also same use the id and schedule the task
- **Maintain certain time limit for all the apps since it is deadline based.**

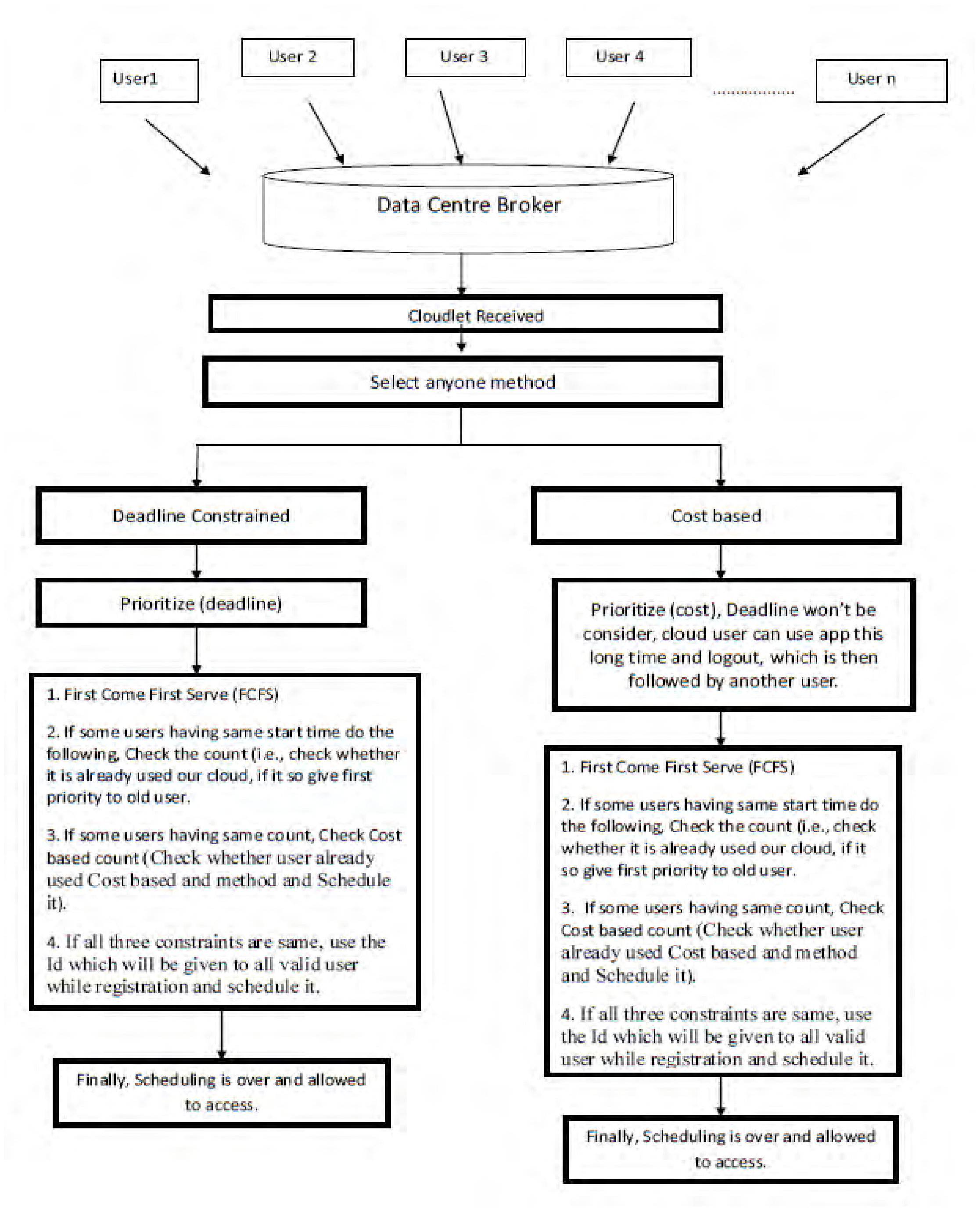

User1
User 2
User 3
User 4
User n
Data Centre Broker
Cloudlet Received
Select anyone method
Deadline Constrained
Cost based
Prioritize (deadline)
Prioritize (cost), Deadline won't be consider, cloud user can use app this long time and logout, which is then followed by another user.
1. First Come First Serve (FCFS)
2. If some users having same start time do the following, Check the count (i.e., check whether it is already used our cloud, if it so give first priority to old user.
3. If some users having same count, Check Cost based count (Check whether user already used Cost based and method and Schedule it).
4. If all three constraints are same, use the Id which will be given to all valid user while registration and schedule it.
Finally, Scheduling is over and allowed to access.

Figure-5 Proposed Algorithm

ii)For Cost Based,

For each user request do

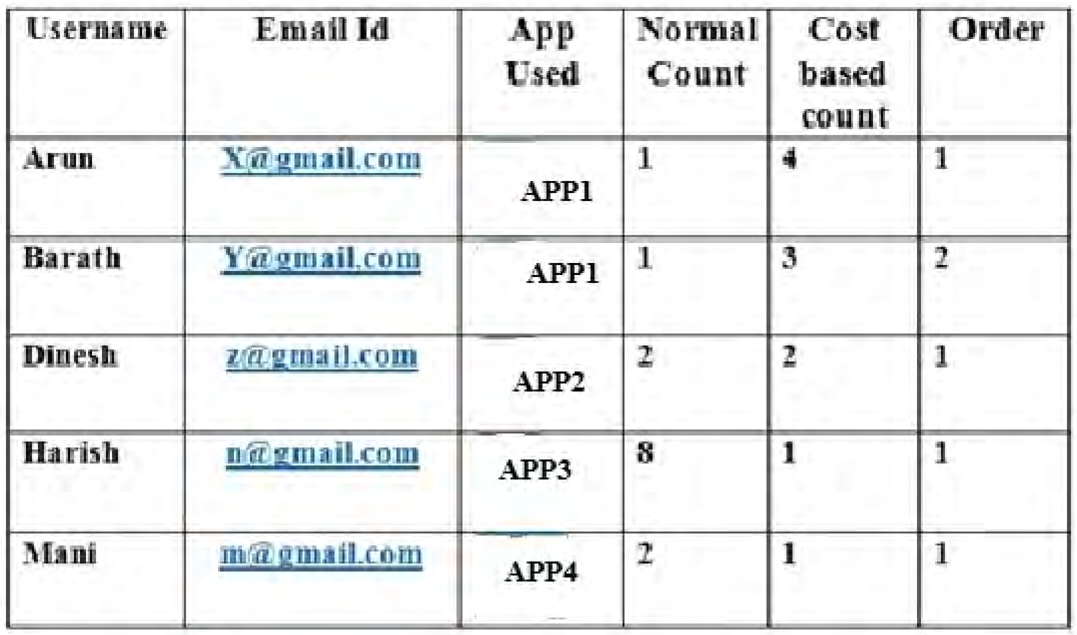
- (i)Compute Start time of each user(first come first serve)
  (ii) Select user with minimum start time and schedule the task.
  (iii) Update the application status.
- (i)If some users having same start time do the following , Check the count(i.e., check whether user already used the cloud)
  (ii) Select user with maximum count and schedule the task.
  (iii) Update the application status.
- (i)if some users having same count, use cost based count(Check he used cost based method)
  (ii) Select user with maximum cost based count and schedules the task.

(iii) Update the application status.

- If cost based count also same use the id and schedule the task
- No need to maintain any time limit, since it is a cost based.

Experimental Results:

CloudSim (Simulator) is used to verify the correctness of proposed algorithm[7,8]. The full method is processed and experiment results are shown below,



Username	Email Id	App Used	Normal Count	Cost based count	Order
Arun	X@gmail.com	APP1	1	4	1
Barath	Y@gmail.com	APP1	1	3	2
Dinesh	z@gmail.com	APP2	2	2	1
Harish	n@gmail.com	APP3	8	1	1
Mani	m@gmail.com	APP4	2	1	1

III.Conclusion and Future work

It is observed that the proposed algorithm improves profit for service providers as compared to other algorithms and also we found that it is very efficient to user. The results improve with the increase in user count.

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

The future work may consider other factors like type of task, task length could be taken into account for proper scheduling of tasks.

References:

[1] ACET, University of Reading Presented by, Prof Mark Baker http://acet.rdg.ac.uk/~mab

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

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

[4] 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.

[5] 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.

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

[7] 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.

[8] Monika Choudhary, Sateesh Kumar Peddoju ‘A Dynamic Optimization Algorithm for Task Scheduling in Cloud Environment’ Vol. 2, Issue 3, May-Jun 2012, pp.2564-2568