IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 10, Issue 5 (Mar. - Apr. 2013), PP 01-05
www.iosrjournals.org

Optimization of FCFS Based Resource Provisioning Algorithm for Cloud Computing

Aditya Marphatia[1], Aditi Muhnot[2], Tanveer Sachdeva[3], Esha Shukla[4], Prof. Lakshmi Kurup[5]

[1]*(Department of Computer Engineering, D.J. Sanghvi College of Engineering/ University of Mumbai, India)*
[2]*(Department of Computer Engineering, D.J. Sanghvi College of Engineering/ University of Mumbai, India)*
[3]*(Department of Computer Engineering, D.J. Sanghvi College of Engineering/ University of Mumbai, India)*
[4]*(Department of Computer Engineering, D.J. Sanghvi College of Engineering/ University of Mumbai, India)*
[5]*(Asst. Professor, Department of Computer Engineering, D.J. Sanghvi College of Engineering/ University of Mumbai, India)*

Abstract: *In our project, we propose an optimized version of the FCFS scheduling algorithm to address the 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 (FCFS manner). Resource selection is done on the basis of task constraints using a greedy approach. The proposed model will be implemented and tested on simulation toolkit. We intend to create a module depicting the normal FCFS algorithm in comparison to our optimized version algorithm for resource provisioning in the cloud.*

Keywords - *Cloud computing, FCFS, module, resource, scheduling*

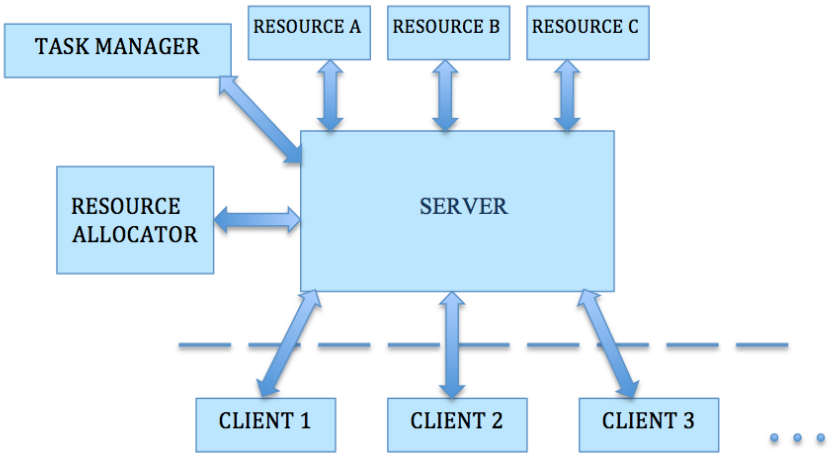
I. INTRODUCTION

Cloud computing is a recent advancement wherein IT infrastructure and applications are provided as 'services' to end-users under a usage-based payment model. It can leverage virtualized services even on the fly based on requirements (workload patterns and QoS) varying with time. Clouds aim to power the next-generation data centers as the enabling platform for dynamic and flexible application provisioning. 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. 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 [1]. But the priority based scheduling may lead to long waiting time for low priority tasks. FCFS algorithm has a major disadvantage that shorter processes may have to wait for a long time until the maximum burst time process finishes execution.

This paper is divided in to the following sections; Architecture, Algorithm, Experiment, Results, Conclusion, Future Scope and References. We begin by describing our system architecture. This is followed by an explanation of our scheduling algorithm. Our experiment is explained thereafter along with the corresponding results. We then conclude the paper with some facts and the scope for further development.

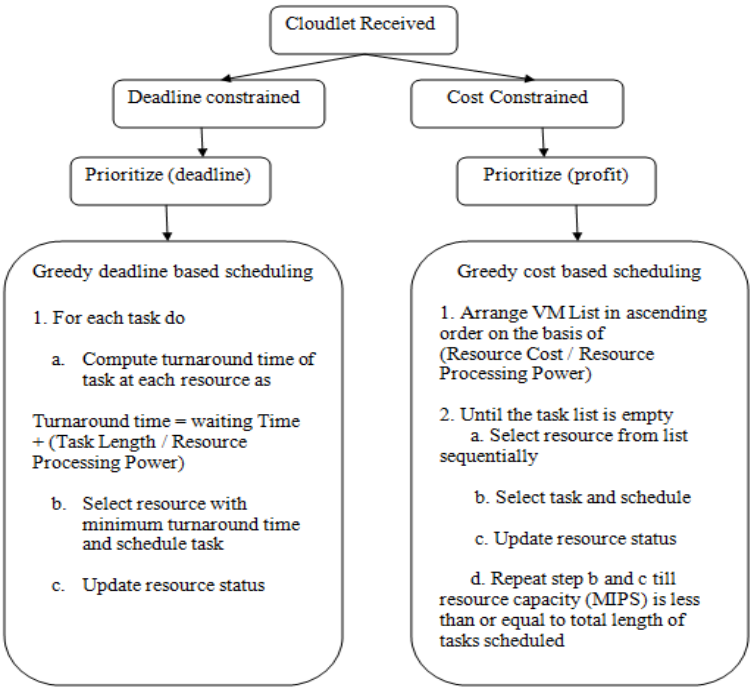
II. ARCHITECTURE

The architecture follows a client server model. As shown in the diagram, each resource is allocated as per requested by various clients through the server. The clients submit their requests to the server that in turn uses the resource allocator to allocate the request. The resource allocator is the program which allocates and de-allocates resources from the clients. It refers to the task manager through the server to study the client usage of the various resources and make an optimum decision.



III. ALGORITHM

The algorithm is an optimized version of FCFS. In case of FCFS if the required resource is unavailable, then the system simply waits for availability whereas our algorithm would give the resource in parts or simply put the request in a wait queue and see if the next request can be serviced. It follows a dynamic allocation towards deadline constraint or cost constraint depending on current usage. It then proceeds to allocate data to requests based on whichever category the request would fit into. For deadline constraint, if a request is below the threshold value, it is immediately serviced, whereas if it is above the value it is considered for cost constraint based allocation [2]. Within cost constraint allocation, among simultaneous requests the request that provides with most cost efficiency is allocated first and so on.



IV. EXPERIMENT

To better understand the cloud environment and the resource allocation strategy we have implemented, let us go over the sequence of operations that take place:

1. A Server Client virtual connection is created. This connection is used to communicate between the server and all its clients.
2. The client then sends request messages to the server asking for the number of resources it requires. Many clients can ask for the same resource or for different resources [3].

3. The server receives these requests from the virtual machine and passes it to the Resource allocator. The Allocator runs the proposed algorithm (or FCFS) either by deadline or cost and then allocates the said resources to the clients that have requested it [8].
4. The Task manager will show the different parameters i.e. CPU utilization or Memory usage by which we can prove our algorithm is logically better at resource provisioning than the usual FCFS based algorithm [4].

To verify the difference in performance between the proposed algorithm and the FCFS based algorithm, we have considered the following test case:

Given:

The number of Virtual Machines requesting for resources is equal to 3, namely A, B and C.
The Resource R has 30 units that can be allocated.
Threshold value for R is 80% of total* i.e. 24 units.
Proposed Algorithm conditions:
Primary decision criterion: Resource available
Secondary decision criterion: Time for which resource is required
Maximum resource grant to one VM = 10 units
Maximum time for which resource can be allocated (if primary condition is violated) = 20 seconds
Testing condition:

i. Initially, A asks for 10 units of R for 40 seconds.
ii. After 5 seconds, B asks for 15 units of R for 20 seconds.
iii. After 15 seconds, C asks for 6 units of R for 20 seconds.

For FCFS:

i. 10 units of Resource R are allocated to A for 40 seconds (remaining=14 units)
ii. Resource R cannot be allocated to B, and B is put on waitlist till resources become available.
iii. After 20 seconds of no activity for remaining resources, C is given 6 units of R for 20 seconds.
iv. After 40 seconds from start, 15 units of R are re-allocated to B.

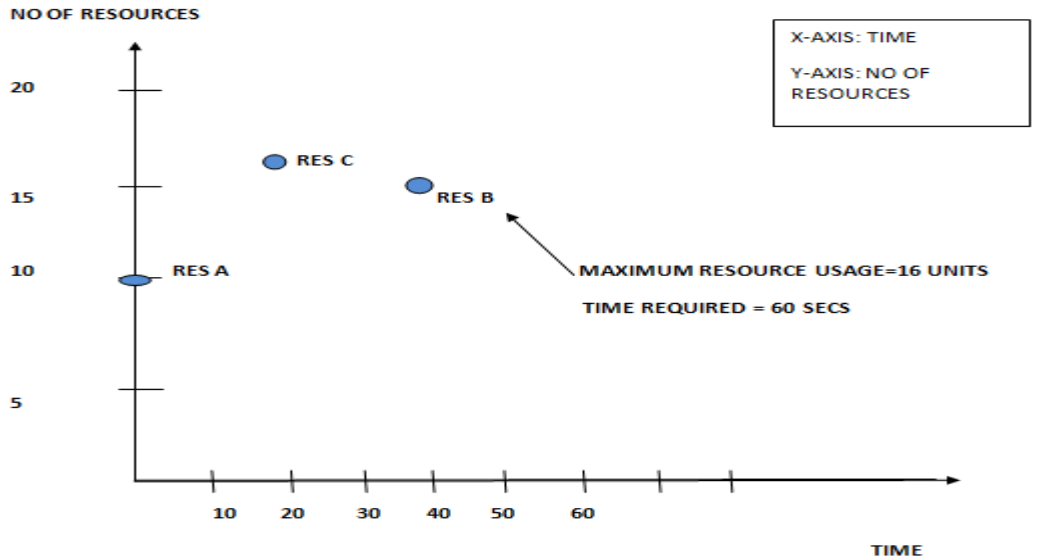
For Proposed Algorithm:

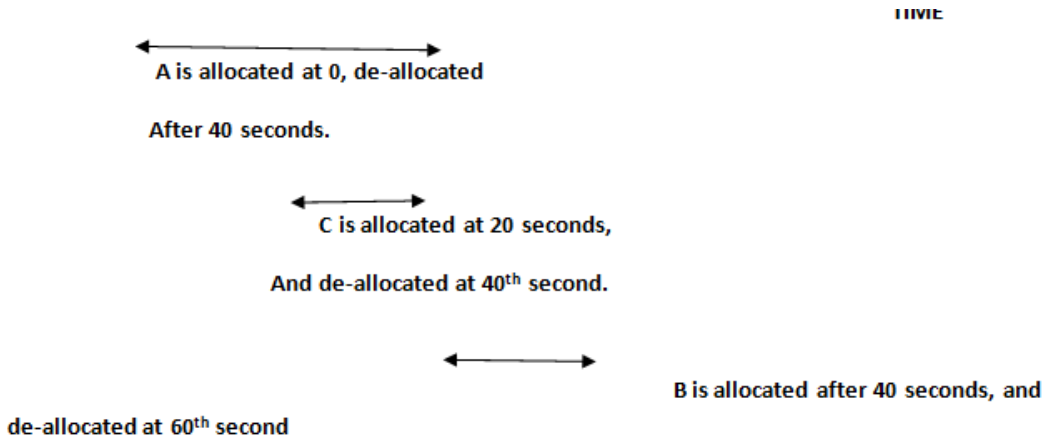
i. 10 units of Resource R are allocated to A for 40 seconds (remaining=14)
ii. As B is violating both testing conditions, half of the required resources i.e. 7 units are allocated to B for 40 seconds.
C will be allocated 6 out of the remaining 7 units of resource R for 20 seconds.

V. RESULTS

For FCFS:

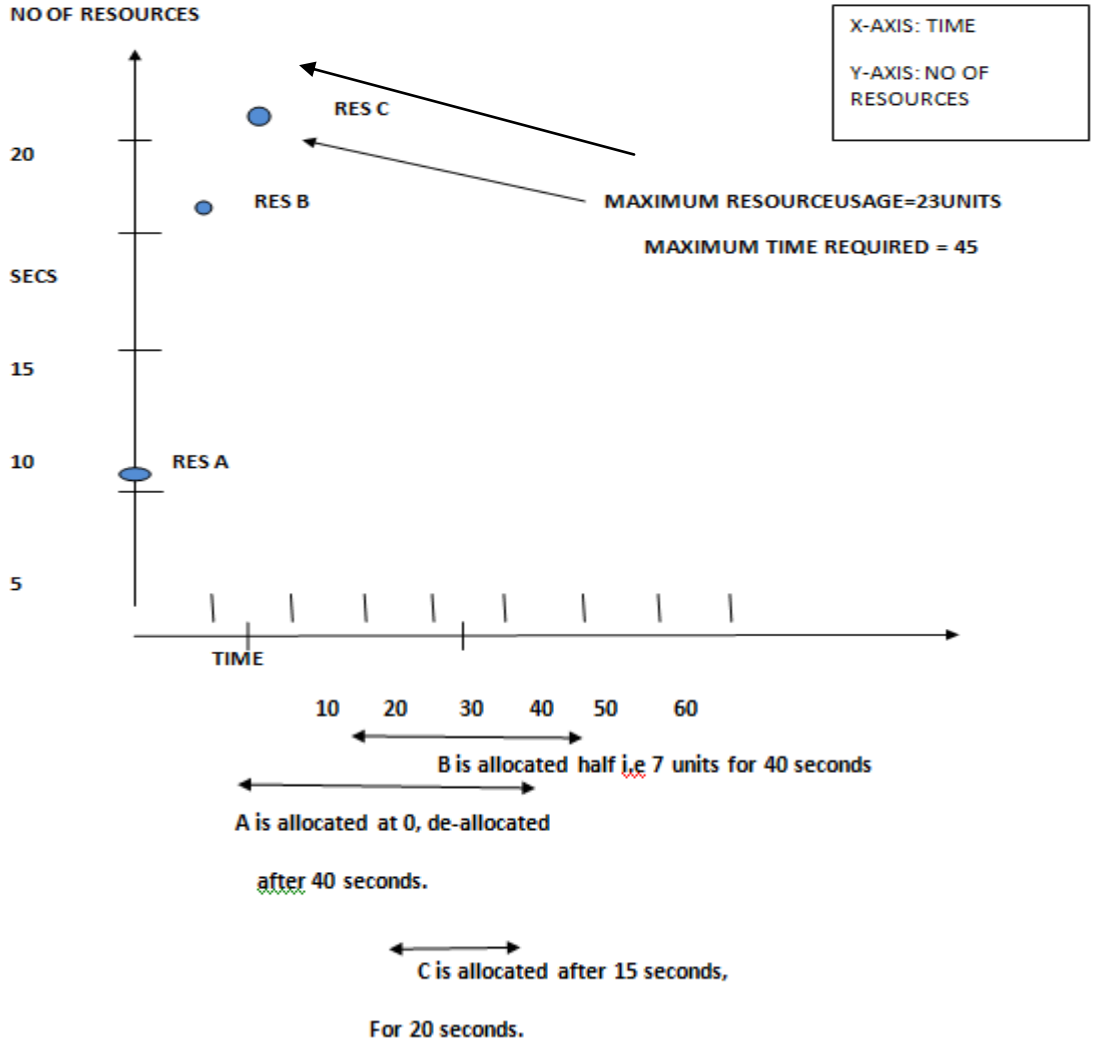
Total time required by test case for completion is 60 seconds and maximum resource usage is 16 units of resources.





For Proposed Algorithm:

Total time required by test case to reach conclusion is 45 seconds and maximum resource usage is 23 units of resources.



VI. CONCLUSION

We have provided a graphic user interface for our clients to log in and submit their requests for resources for a particular period of time. Our proposed algorithm derives its data from the task manager and optimizes the FCFS algorithm by considering deadline and cost constraints. Our algorithm dynamically

allocates resources depending upon the specification of each user request. Upon completion of the timer based allocation period the said resources are revoked from the client and the memory is de-allocated [6].

This enables optimized memory usage and enhances system performance from both server and client perspective. The adjacent experiment results exhibit the user-friendly interface and system performance under user requests.

VII. RESULTS

Our project can act as the foundation in both academia and practical implementation through an application. Future work will entail a lot of data and software on a cloud environment. As a central server, resource allocation to its clients will be one of the major things resource providers need to think about. We believe using our algorithm on a large scale optimizes resource allocation to an extent not many general algorithms can provide without suffering huge calculation costs [5].

Although, we have developed an algorithm that is based on priority as well classification and has different parameters to allocate resources such as deadline or cost, there can be improvements with respect to other parameters as well. As we have used virtual resources, we can even use actual physical resources for allocation.

Moreover, we believe that the OS virtualization setup can be extended to different computers and make a distributed cloud environment with slight modifications [7]. Additionally, certain performance checking criteria can be added to the original set up to demonstrate an algorithm which can easily outperform the FCFS algorithm in all respects.

REFERENCES

JOURNAL PAPERS:

[1] IEEE/ACM Int. Conf. On Grid Computing (Grid 2008)

[2] 85-94, 2008.[11]Grosu, D., Das, A., "Auction-Based Resource Allocation Protocols In Grids", 16th Iasted International Conference On Parallel And Distributed Computing And Systems, 2004.

[3] Izakian, H., Ladani, B.T., Zamanifar, K., Abraham, A., Snasel, V., "A Continuous Double Auction Method For Resource Allocation In Computational Grids", IEEE Symposium On Computational Intelligence In Scheduling, 2009, Pp. 29-35, 2009.

[4] H. Izakian, A.Abraham, B. Tork Ladani. An Auction For Resource Allocation In Computational Grids .Future Gneration Computer Systems Vol 26 ,Pp 228–, 2011.

[5] Xiaohong Wu At La. Cloud Computing Resourceallocation Mechanism Research Based On Reverseauction. Energy Procedia Vol 13, 2011 ,Pp 736–741

[6] I. Fujiwara. Applying Double-Sided Combinational Auctions To Resource Allocation In Cloud Computing, IEEE 10th Annual International Symposium On Applications And The Internet, 2010.

[8] Adabi, S., Movaghar, A., Rahmani, A., M., Beigy, H.,Market-Based Grid Resource Allocation Using New Negotiation Model", Journal Of Network Andcomputer Applications, 2012.