ISSN:0975-9646

A Comparative Study on Various Scheduling Algorithms of Cloud Computing

NUPUR AGRAHARI[1] Er.VIKASH[2]

[1,2]Department Of Information Technology Greater Noida Institute of Technology Greater Noida, India

Abstract- **Cloud computing is an extensively trending topic in the present era. The study here attempts to explain cloud computing in relatively simple terms in brief and further to develop an algorithm that helps in efficient task scheduling in such a manner that is beneficial to both the users as well as the cloud service providers. As we move further we finally compare the efficiencies of the existing algorithms that are currently in use and the proposed algorithm.**

Key words: **Greedy Resource Allocation Algorithm**

I. INTRODUCTION

Cloud computing is a subscription-based service where you can obtain networked storage space and computer resources. One way to think of cloud computing is to consider your experience with email. Your email client, if it is Yahoo!, Gmail, Hotmail, and so on, takes care of housing all of the hardware and software necessary to support your personal email account. When you want to access your email you open your web browser, go to the email client, and login, all that you need is internet access. Our main goal here is *"To design an optimal scheduling algorithm that minimize the cost and task completion time experienced by users for the execution of their tasks while simultaneously utilizing the cloud resources to their maximum."* .We must take the properties of the cloud environment into account to make cloud services and cloud-oriented applications efficient. By efficient, we mean appropriate resources are allocated at a right time to a right application, so that applications can utilize the resources effectively. In other words, we want to minimize the amount of resources for an application to maintain a desirable level of service quality, or maximize throughput (or minimize job completion time) of an application. The solution is the proposed algorithm which minimizes the turnaround time and cost of each job individually to minimize the average turnaround time and cost of all submitted tasks in a time slot respectively.

II. LITERATURE REVIEW

A four-layer architecture for cloud computing is shown in Figure 2.1

The fabric layer contains the raw hardware level resources, such as compute resources, storage resources, and network resources. On the unified resource layer, resources are virtualized so that they can be exposed and made available to upper layer and end users as integrated resources. The platform layer adds on a collection of specialized tools, middleware and services on top of the unified resources to provide a development and deployment platform. The application layer includes the applications that would run in the clouds.

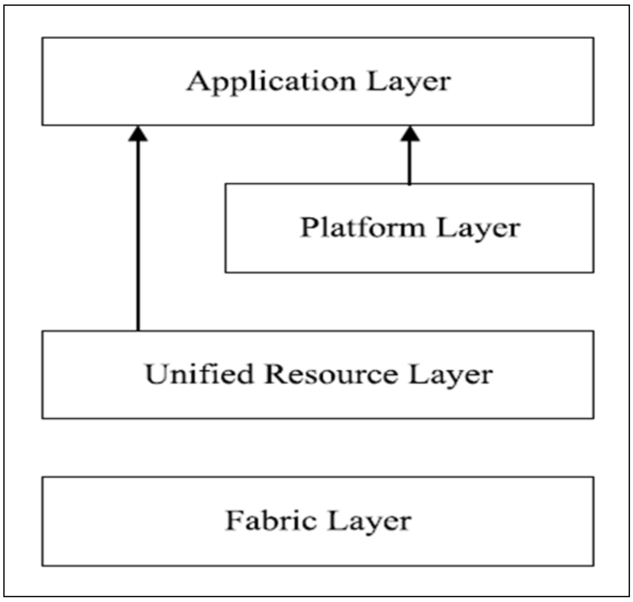

Figure 0.1 Cloud Architecture

A SURVEY ON RESOURCE ALLOCATION STRATERGIES IN CLOUD COMPUTING

Cloud computing emerges as a new computing paradigm which aims to provide reliable, customized and QoS (Quality of Service) guaranteed computing dynamic environments for end-users. Distributed processing, parallel processing and grid computing together emerged as cloud computing. The basic principle of cloud computing is that user data is not stored locally but is stored in the data centre of internet. The companies which provide cloud computing service could manage and maintain the operation of these data centres. The users can access the stored data at any time by using Application Programming Interface (API) provided by cloud providers through any terminal equipment connected to the internet.

Not only are storage services provided but also hardware and software services are available to the general public and business markets. The services provided by service providers can be everything, from the infrastructure, platform or software resources. Each such service is respectively called Infrastructure as a Service (IaaS), Platform as a Service (PaaS) or Software as a Service (SaaS).

GREEDY ALGORITHM

The Greedy algorithm is the default algorithm used for scheduling of Virtual Machines in Eucalyptus. The Greedy algorithm is very simple and straightforward. As a matter of fact, it was the only scheduling policy which was in use for a longtime. Only after the cloud started evolving, more complex scheduling policies came into effect.

The greedy algorithm uses the first node that it finds with suitable resources for running the VM that is to be allocated. The first node that is identified is allocated the VM. This means that the greedy algorithm exhausts a node before it goes onto the next node.

As an example, if there are 3 nodes and the first node's usage is 40% while the other two are under loaded and if there are two VMs to be allocated, then both are allocated to the first node which might result in the increase of its usage to 90% while the other two nodes will still remain under loaded.

As obviously seen, the main advantage of the Greedy algorithm is its simplicity. It is both simple to implement and also the allocation of VMs do not require any complex processing. The major drawback would be the low utilization of the available resources. As illustrated in the example above, even if there are under loaded nodes, an overloading of a node might result.

III. APPLICATION SCENARIO

CloudSim SIMULATOR

CloudSim is a generalized and extensible simulation framework that enables seamless modelling, simulation, and experimentation of emerging Cloud computing infrastructures and application services. By using CloudSim, researchers and industry-based developers can focus on specific system design issues that they want to investigate, without getting concerned about the low level details related to Cloud-based infrastructures and services. There are various compelling features of CloudSim that have promoted the development of new resource allocation policies and scheduling algorithms for Cloud.

ARCHITECTURE

Physical Cloud resources along with core middleware capabilities form the basis for delivering IaaS. The user-level middleware aims at providing PaaS capabilities. The top layer focuses on application services (SaaS) by making use of services provided by the lower layer services. PaaS/SaaS services are often developed and provided by 3rd party service providers, who are different from IaaS providers. The layered design of service-oriented Cloud computing architecture is shown in figure 3.1

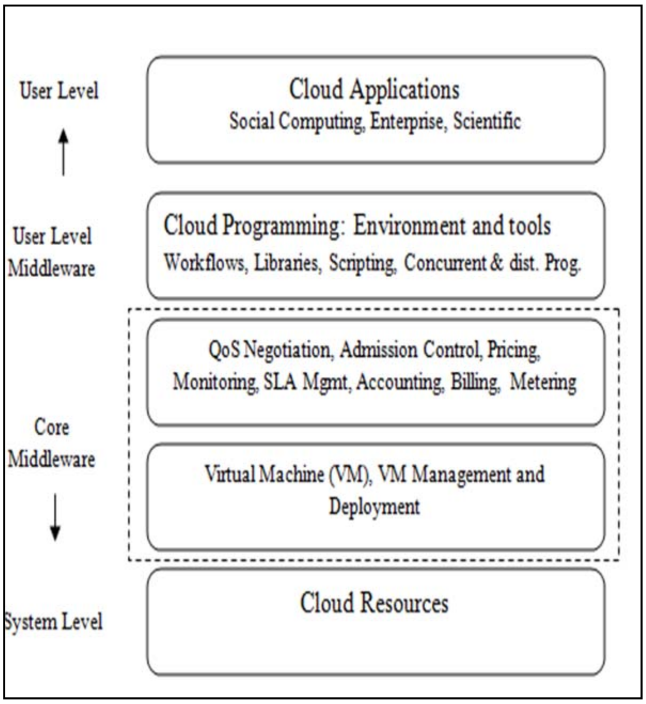

Figure 3.1 CloudSim Architecture

EXISTING SCHEDULING ALGORITHMS IN EUCALYPTUS

Scheduling in Eucalyptus determines the method by which Virtual Machines are allocated to the nodes. This is done to balance the load on all the nodes effectively and to achieve a target quality of service. The need for a good scheduling algorithm arises from the requirement for it to perform multitasking and multiplexing.

The scheduling algorithm in Eucalyptus is concerned mainly with:

Throughput- number of VMs that are successfully allocated per time unit.

Response time-amount of time it takes from when a request was submitted until the first response is produced.

Fairness /Waiting Time – All the requests for an allocation of a node should be treated in the same manner without any bias.

In practice, these goals often conflict and thus a scheduler will implement a suitable compromise. Preference is given to anyone of the above mentioned concerns depending upon the user's needs and objectives. A scheduling policy can be chosen by changing the value of SCHEDPOLICY in eucalyptus Conf file. The various existing algorithms along with their limitations are listed below.

IV. THE ALGORITHM

Here we have compared various algorithms which are used for scheduling before we come to the final algorithm. The algorithm provided below has been developed with by comparing various scheduling algorithms shown in Table 1.

THE ALGORITHM

```
Inputs: Virtual Machine
Outputs: outer ResID;B
Begin
setresid1=sleepresid1=-1, execute=0;
//find the best 'resource' on which to run the instance
For each resource in Resource Cache & not done
do
if (resource is in(Suspended State or Waking State)and
resid1==-1)then
find remaining Capacity of resource and check
if remaining resource capacity>0then
resid1=resurceid.execute++
end
r
endif(resource is in sleeping State)and resid1==-1
)then find remaining Capacity of resource and
Check
If remaining resource capacity>0then sleep
resid1=resurceid1;
end
end
end
if(resid1==-1&sleepresid1==-1)then
outResID=-1,returnoutResId
end
if(resid1==-1)then get resource from resource cache
having resid

outResID=resid
endif(sleep resid1==-1)then
get resource from resource  cache  having sleep resid out
ResID=sleepreid1
endif resource is sleeps ate then
power Up resource
end return out ResIDend
```

COMPARISON OF VARIOUS SCHEDULING ALGORITHMS:

Scheduling Algorithm	Scheduling Method	Scheduling Parameter	Scheduling Factor	Findings	Environments
Resource Aware Scheduling algorithm (RASA)	chMode	Make span	Occupied Task	It is used to do reduce make span	Cloud Environment
Reliable scheduling distributed in cloud computing	chMode	Accessing Time	Occupied Task	It is used to enhance Processing time. It is efficient for load balancing.	Cloud Environment
Optimal Model for Priority based Service Scheduling Policy for Cloud Computing	chMode	Utility Of Services Service Request time	Array of work flow instances	High QoS lightweight Throughtput.	Cloud Environment
priority based Scheduling Algorithm in Cloud Computing	Dependency mode	Priority to each queue	Array of Job Queue	Less finish time	Cloud Environment
Extended Max-Min Scheduling Using Petri Net and Load Balancing	chMode	Load Balancing time	Occupied Task	It is used for client load balancing.	Cloud Environment
Optimistic Differentiated Service Job Scheduling System for Cloud Computing	dependency mode	Utility of Services	Single Job with Multipleuser.	It Achieve QoS requirements of the cloud computing user and the maximum profits of the cloud computing service provider are achieved.	Cloud Environment
optimized cost based task scheduling in Cloud Computing	chMode	Test , Performance	Scheduled Task	It ensure both source cost and computation performance improves the computation communication ratio.	Cloud Environment
Gang Scheduling In Cloud Computing	chMode	Performance , cost	Work Flow with change number of job	The application migration and starvation handling.	Cloud Environment
Greedy algorithm	chMode	Prioritization	Resource Allocation	1.Resource discovering and filtering. 2. Resource selection according to certain objectives. 3.Task Submission.	Cloud Environment

V. RESULT

The CloudSim toolkit is used to simulate heterogeneous resource environment and the communication environment. CloudSim simulator is used to verify the results. 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 scheduler submits these jobs on available resources according to these algorithms. All parameters are varied in a similar fashion for judging the performance of the two algorithms.

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

Table 5.1 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 below table..

Virtual Machines	Virtual Machine 1	Virtual Machine 2
RAM(MB)	5024	5024
Processing Power(MIPS)	22000	11000
Processing Element(CPU)	1	1

Table 5.2 Configuration of VMs

Performance with Respect to Execution Cost

The cost based tasks use greedy approach to minimize the execution cost of individual tasks by selecting the appropriate resource. The tasks execution using the proposed algorithm results in a significant improvement in cost over the sequential allotment as shown in Table 4. The improvement in cost increases with the increase in number of cloudlets.

Table 5.3 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

The resulting graph comparing the resultant completion time is shown in figure 5.1 with task on x axis and execution cost on y-axis.

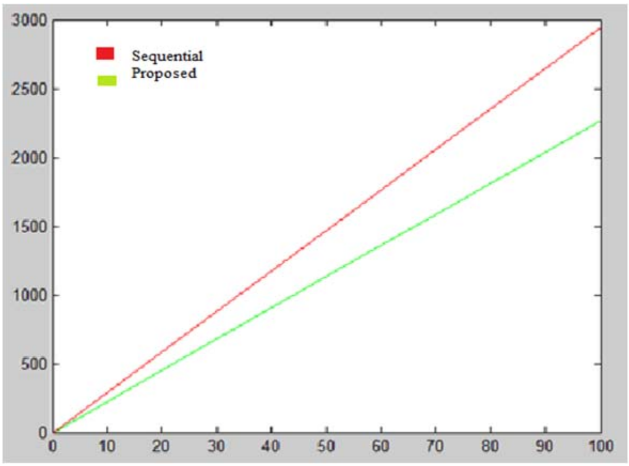

Figure 0.1 Analysis of Execution Cost

Performance with Respect to Deadline

The experiment is performed repeatedly increasing the number of cloudlets in each trial. It is evident from the results that proposed algorithm gives better completion time of job in comparison to the sequential approach.

Table 1.4 Comparison of Completion Time

No. Of Cloudlets	Proposed Algorithm	Sequential Approach
25	52.97	58.97
50	164.66	181.62
75	334.49	399.90
100	584.68	654.03
125	910.04	997.99
150	1298.50	1439.75

The resulting graph comparing the resultant completion time is as shown in figure 5.2 with task on x axis and completion time on y-axis.

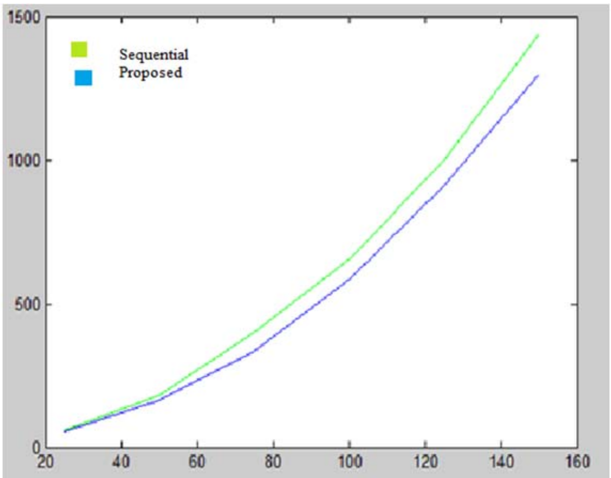

Figure 0.2 Analysis of Completion Time

VI. CONCLUSION AND FORMATTING

Conclusion

Scheduling is an important aspect for achieving best results with available resources in the cloud computing system. In the cloud environment the jobs are executed on different resources which are geographically distributed. The key challenge is to schedule tasks in such a way that result is beneficial both for the user and cloud service provider. This happens when tasks are executed in accordance to their QoS specifications with minimum cost, minimum time and efficient utilization of resource capacity. This problem is addressed by proposing and implementing a scheduling algorithm which schedules incoming tasks on the basis of their cost and time requirements.

The following conclusions can be drawn from the present study:

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.

Suggestions for Future Work

There is substantial scope for improvement in the scheduling approaches used in cloud environment. 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. 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

[8] 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.

[9] T. J. Velte, *et al.*, *Cloud Computing: A Practical Approach*: McGraw-Hill Osborne Media, 2009.

[10] V. Hamscher, *et al.*, "Evaluation of job-scheduling strategies for grid computing," *Grid Computing—GRID 2000,* pp. 191-202, 2000.

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

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

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

[14] 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.

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

[16] 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.

[17] R. N. Calheiros, *et al.*, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience,* vol. 41, pp. 23-50, 2011.