

Task Scheduling Algorithm in Cloud Computing based on Power Factor

Sunita Sharma¹, Nagendra Kumar²P.G. Student, Department of Computer Engineering, Shri Ram Institute of Science & Technology, JBP, M.P, India¹Assistant Professor, Department of Computer Engineering, Shri Ram Institute of Science & Technology, JBP,
M.P, India²

ABSTRACT: A very efficient computing environment is provided by cloud computing where the customers or several tenants are in need of multiple resources to be provided as a service over the internet. The challenge before the cloud service provider is, how efficiently and effectively the underlying computing resources like virtual machines, network, storage units, and bandwidth etc. should be managed so that no computing device is in under-utilization or over-utilization state in a dynamic environment. A good task scheduling technique is always required for the dynamic allocation of the task to avoid such a situation.

Through this paper we are going to present the new Algorithm based on task scheduling technique, which will distribute the load effectively among the virtual machine so that the overall response time (QoS) should be minimal. A comparison of this proposed Algorithm of task scheduling technique is performed on CloudSim simulator which shows that, this will outperform the existing techniques like FCFS, SJF and Genetic Model techniques.

KEYWORDS: Cloud Computing, Task Scheduling, FCFS, SJF, Genetic Algorithm, QoS.

I. INTRODUCTION

When we have the Internet as cloud, then we are representing abstraction, which is one of the most important and essential property of cloud computing. In the Internet, which is called as cloud, resources are pooled and partitioned as needed and communications are based on some standards [1]. A new type of computer model i.e. Cloud provides the user with convenient and ever-demanding access of network for various computing resources over the Internet. Cloud computing which is a huge distributed computing environment contains a large amount of virtualized computing resources available for individual or an organization. Reason behind development or introduction of cloud computing is to provide the guarantee of Quality of Service (QoS) which is quite challenging. Scheduling of job and maintaining load is becoming a main issue in cloud environment. This can be achieved by adopting appropriate task scheduling algorithm.

Cloud computing basic component: In this section, we will discuss the basic components on which cloud computing deployed. These components consist of a wide range of services that we can use all over the internet. Here we discuss some important component:

Virtualization: It plays an important role in deploying the cloud. It is the strategic component in the cloud, which allows the physical resources by multiple consumers. It creates the virtual instance of resource or device such as operating system, servers, network resources and storage devices wherein the framework utilize the resources into more than one execution environment.

Multi-tenancy: Multi-tenant environment can have multiple customers or users who does not see or share each other's data but can share resource or application in an execution environment, even if they may not belong to the same organization. Multi-tenancy results the optimal utilization of hardware and data storage mechanism. This thesis address issues related to this area.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

Cloud storage: It is a component, which maintained, managed, and backed up remotely and it made available over the network where the users can access data.

The hypervisor: The So called virtual machine monitor or manager is a key module of virtualization. It allows multiple Virtual Machines (VMs) to run on a single hardware host. It manages and monitors the various operating systems, which run in a shared physical system.

II. RELATED WORK

A. Scheduling in Cloud Computing: The main objective of task scheduling algorithms is to maximize the resource utilization in cloud environment. Minimizing the waiting time of resources in cloud data centre is the main goal of scheduling. A good scheduling algorithm always yields good system performance. In the cloud there are numerous and distinct resources available. The cost of performing tasks in cloud depends on which resources are being used so the scheduling in a cloud environment is different from the traditional scheduling. In a cloud computing environment task scheduling is a biggest and challenging issue.

A Cloud provider first constructs a computing system called cloud. In this we have several virtual machines interconnected and the provider processes the task of the users. "Cloud computing is not a well behaved model for providing wanted, user- required, flexible access to a shared pool of configurable computing resources that can be quickly provided and released with low care effort or service are going to study the divisible task scheduling of high performance computing algorithms" [2]. Cloud computing environment where multiple virtual machines (VMs) can share physical resources (CPU, memory, and bandwidth) on a single physical host and multiple VMs can share the bandwidth of a data center by using network virtualization. Because many users and applications essentially share system resources, a proper task-scheduling scheme is difficult to resource utilization and system performance. Many system parameters, such as processor power, memory space, and network bandwidth, affect the efficiency of task scheduling.

In addition, difference in computing sources in different nodes adds to the complexity of task scheduling. Furthermore, frequent data exchange among nodes, hosts, and clusters in data-intensive cloud applications makes the task-scheduling procedure extremely complicated. Most of these methods focus on allocating CPU and memory resources to various cloud-computing tasks, assuming that all physical nodes and VMs have unlimited network bandwidth. The multi-dimensional task scheduling algorithm is based on the availability of CPU, memory, and VMs. This algorithm considers the limitation of resources, and provides resources according to task needs and resource loads. Computing resources like CPU, memory and bandwidth are used by many users, so it is little difficult to construct an efficient task scheduling algorithm. The efficiency of the algorithm is affected by many things like the processor power, speed, space and memory [3]. On behalf of this the heterogeneity of computing sources also causes damage to the efficiency of algorithm.

Generally, task scheduling is the main process in infrastructure as a service model. While scheduling the task we consider virtual machines as scheduling machines. The main aim of task scheduling algorithms in cloud environment is to maximize the utilization of resources, efficiency and to reduce their task execution time.

Task management is an umbrella activity comprising of different stages of resources and workloads from workload submission to workload execution. Task Scheduling in Cloud includes two stages:

- i) Resource provisioning and
- ii) Task scheduling.

Task Scheduling is defined to be the stage to identify adequate resources for a given task based on QoS requirements described by cloud consumers whereas resource scheduling is mapping and execution of cloud consumer workloads based on selected resources through resource provisioning as shown in Fig.1.2. Firstly, cloud consumer submits request for task/workload execution in the form of workload details. Based on these details broker (resource provisionor) finds the suitable resource(s) for a given task and determines the feasibility of provisioning of resources based on QoS requirements [4]. Broker sends requests to resource scheduler for scheduling after successful provisioning of resources. Other responsibilities of broker include: release of extra resources to resource pool, contains information of provisioned resources and monitor performance to add or remove resources. After resource provisioning, resource scheduling is done in second stage.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

All the provisioned resources are kept in resource queue while other remaining resources are in resource pool [5]. Submitted tasks are processed in task queue. In this stage, scheduling agent maps the provisioned resources to given tasks, execute the tasks and release the resources back to resources pool after successful completion of tasks. Based on QoS requirements, scheduling of resources for adequate task is a challenging issue. For an efficient scheduling of resources, it is necessary to consider the QoS requirements [6].

Effective task scheduling algorithms reduces execution cost, execution time, energy consumption and considering other QoS requirements like reliability, security, availability and scalability.

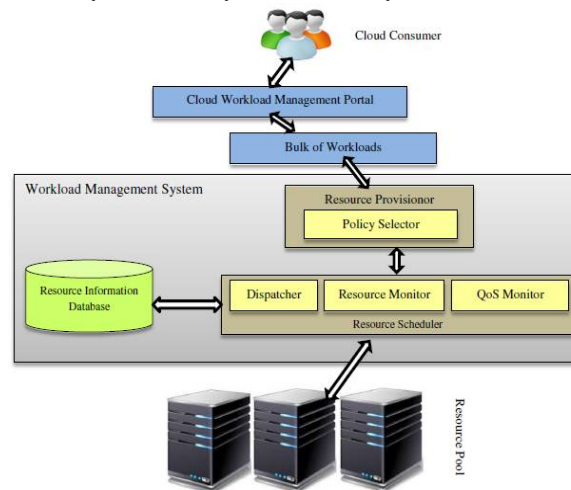


Fig. 1 Scheduling Architecture in Cloud Computing.

In cloud environment, cloud consumer and cloud provider are two parties. Cloud consumer submits tasks while cloud provider provides resources for execution of tasks. Both the parties have different requirements: provider wants to earn as much profits as possible with lowest investment and maximize utilization of resources while consumer wants to execute their tasks with minimum cost and execution time. However, executing number of tasks on one resource will create interference among tasks which leads to poor performance and reduces customer satisfaction. To maintain the service quality, providers reject the requests that result in unpredictable environment [7]. Providers also consider unpredictable resources for scheduling and execution of the tasks. Scheduling of tasks to proper resources becomes more challenging because both user and providers are not willing to share information with each other. The challenges of resource scheduling include dispersion, uncertainty and heterogeneity of resources that are not resolved with traditional algorithms in cloud environment. Therefore, there is a need to execute cloud tasks in an efficient way by taking care of these properties of the cloud environment.

B. Types Scheduling In Cloud Computing

Scheduling methods in cloud environment generally have been divided into three groups: Resource scheduling, Workflow scheduling and Task scheduling, in which only the task scheduling approaches are focused in this thesis. The entire classification is portrayed in Fig. 2. Resource scheduling performs mapping of virtual resources among physical machines and workflow scheduling is done to schedule workflows constituting an entire job in a suitable order. Task scheduling methods may be centralized or distributed. It can be performed in homogeneous or heterogeneous environment on dependent or independent tasks. In centralized scheduling a single scheduler is there to do all mappings whereas in distributed, scheduling is partitioned among different schedulers. Scheduling methods in distributed environment can be of two types: heuristic and hybrid techniques. Heuristic methods are classified into static as well as dynamic scheduling. Dynamic scheduling can be performed in online mode or batch mode. In static scheduling all the tasks are known a priori to scheduling and they are statically assigned to virtual resources. In dynamic scheduling all the tasks are scheduled instantly, as they arrive in the system. Dynamic scheduling mechanism performs better when

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

compared to static algorithms. But the overhead of dynamic algorithms are high as we want to decide the schedule and update the system information instantly.

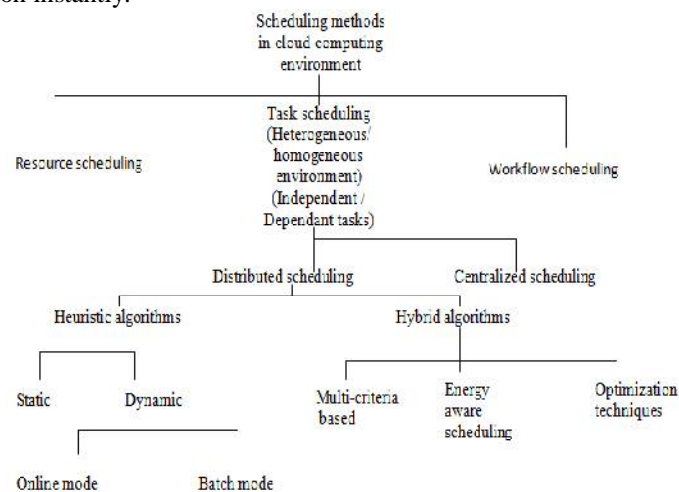


Fig.2 Types of Scheduling in Cloud Computing [8].

First Come First Serve (FCFS), Round robin, Max-Min are the examples of batch mode heuristic algorithms. It initializes the scheduling of tasks after a particular time period and it put all incoming tasks into the queue. In the cloud, scheduling is done in three steps which are discovery and filtering of resource, selection of resource and mapping of the task. Resources available in the system are discovered by Datacenter Broker and it also gathers the related information such as a status of the resource, etc. The second step is the selection of resource among all available resources and this selection is takes place by considering definite parameters of both resource and task. The last step is to map the task to the selected resource [9].

C. Task Scheduling Approaches

There are different scheduling algorithms available for cloud computing. In this paper we will discuss about five scheduling algorithms they are First Come First Serve (FCFS), Round Robin (RR), Genetic algorithm, Match-making algorithm and generalized priority algorithm.

First come first serve (FCFS) Algorithm: FCFS is mainly used for parallel processing, it is selected for incoming task and it is aimed at resource with smallest waiting queue time. The CloudSim toolkit supports FCFS scheduling strategy for internal scheduling of jobs. Allocation of application-specific VMs to host in a cloud based data center is the responsibility of the virtual machine provisioned component. The default policy of implemented by VM provisioning is straightforward policy that allocates a VM to the host in FCFS basis. The disadvantage of FCFS is it is non-preemptive algorithm [10].

The shortest tasks which are at the back of the queue have to wait for the long task at the front finish. Its turn around and response time is low.

Round Robin (RR) Algorithm: The Round Robin algorithm focuses on fairness and on distributing the load equally to all nodes. Each job in a queue has same execution time and it will be executed in turn. The scheduler starts assigning VM to each node and move further for next VM to place in next node. Algorithm is applied for all the nodes until one VM is assigned to each node. Again it goes to the first node repeat this process to the next VM request [10].

The advantage is that it utilizes all the resources in the balanced order. Disadvantage is high power consumption as many nodes are turned on. If four resources have to be run on a single node, all the nodes will be turned on when Round Robin is used. This consumes high power.

Supports RR CloudSim toolkit supports RR scheduling strategy for internal scheduling of jobs. CloudSim toolkit supports RR scheduling strategy for internal scheduling of jobs.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

Genetic Algorithm

Genetic algorithms are stochastic search algorithms based on the mechanism of natural selection strategy. It starts with a set of initial solution, called initial population, and will generate new solution using genetic operators. The genetic algorithm approach computes the impact in advance that it will have on the system after the new VM resource is deployed in the system, by utilizing historical data and current state of the system. It then picks up the solution, which will have the least effect on the system. The advantage of this technique is it can handle a large searching space, applicable to complex objective function and can avoid trapping by local optimum solution. Authors of [11] have developed a cost-based job scheduling algorithm, which provide a multi QoS scheduling in cloud computing environment.

III. PROPOSED WORK

Task scheduling algorithms in cloud data center are responsible for allocating the tasks submitted by the different cloud users to the available cloud resources. The main objective of task scheduling algorithm is to achieve a highly performance computing and the best system throughput. Tasks in cloud computing are submitted to the Data Center Broker by the Users. The Data Center Broker works as a mediator between the cloud users and providers and is responsible for scheduling tasks on virtual machines (VM). Data Center is a virtual Infrastructure for housing resources and consists of a number of Hosts. Data center broker provides a policy for scheduling of submitted tasks. Broker communicates directly with the cloud controller and assigns tasks to Virtual machines in the host of the data center.

Aim of task scheduling algorithms is to minimizing the completion time of tasks and maximizing resource utilization. Proposed algorithm tries to keep all resources busy. It produces a very poor makespan because it doesn't consider the expected execution time. In proposed algorithm main focus is on assigning virtual machines to the tasks, according to power factor. Power factor is calculated by following formula:

$$PF \text{ of } VM_i = VM_i \text{ MIPS} / VM_s \text{ MIPS.}$$

After calculating power factor of each virtual machine they are sorted in ascending order of power factors. After sorting divide virtual machines into two groups- primary and secondary. Primary virtual machines are from 1 to $m/2$ and secondary are from remaining. If submitted task requires power less than or equals to primary group than assigned it to primary otherwise to secondary. So, in proposed method there exist no needs to take resources on rent. Tasks requiring more power will be assigned to secondary group. It improves profit of cloud providers. It also improves makespan and throughput without renting resources.

A) Proposed Algorithm

Input:

- Number n of cloudlets (i.e., tasks).
- Number m of VMs (i.e., resources).

Output:

Mapping Scheme for the requested tasks (cloudlets) on the available resources (VMs).

Algorithm:

Initialize cloudsims package.

// calculate the total processing power of cloudlets

for i=1 to n do

 Define MIPS of the cloudlet

end for

// calculate the total processing power of VMs.

for i=1 to m do

 Define MIPS of the VMs

End for

// calculate the power factor of each VM, and the

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

```

allotment of each VM
for i=1 to m do
    PF of VMi = processing power of VMi / total processing power of all VMS
end for
Sort VM into ascending order of PF
for i= 1 to m/2 do
    PrimaryVM=VMi
end for
for i= m/2+1 to m do
    SecondaryVM=VMi
end for
for i=1 to n
    If expected execution time is high then
        Assign task to PrimaryVM
    else
        Assign task to SecondaryVM
    End if
    VMi allotment =PF *(total requested processing power (TASKS_MIPs).
    find the sum of set (VMi allotment).
end for
Select a set of cloudlets with the sum of their MIPS equal to/or less VMi allotment.
return the set.

```

IV. EXPERIMENTAL RESULTS

This section presents the simulation experiments to demonstrate the effectiveness of the resource allocation strategies developed in this chapter by using CloudSim 3.0.3 tool. The metrics used to measure the performance obtained by resource allocation strategies are mean response time. This section contains the implementation part of this research that means development and simulation process of the proposed algorithm using CloudSim 3.0.3 tool as well as software and hardware requirements for this experiment. The software used for this experiments are Netbeans, Java and CloudSim3.0.3. We have developed the proposed algorithm in Java language. Then CloudSim3.0.3 simulator is used to test the developed algorithm. To test the proposed algorithm, one data center, one scheduler, and 5 virtual machines are used first.

Then the number of virtual machines is increased by 15 and 30 respectively to test the algorithm. We have also developed FCFS and Genetic algorithms in Java using their pseudo codes for the simulation purpose. Because we compare our proposed algorithm with FCFS and Genetic algorithm to check whose performance is better?

There are different sizes virtual machines are used for this evaluation. Table 1 shows the performance comparison between the proposed algorithm, FCFS, and MaxMin with respect to makespan using different workflows. Figure below shows the graphical representation of the results where VM = 10.

Data Centre Size	FCFS ALGORITHM(MS)	GENETIC ALGORITHM(MS)	PROPOSED ALGORITHM(MS)
VM=10.	228.94	198.5	94.73
VM=20	241.49	202.37	98.36
VM=50	298.32	217.3	115.32

Table 1 Makespan Comparisons.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

Results of above evaluations show that proposed algorithm completes tasks execution with lower makespan and higher performance as compared to scheduling algorithms Genetic and FCFS. Performance of proposed algorithm is better than Genetic algorithm and FCFS for 10, 20 and 30 virtual machines. Results shows that proposed algorithm behaves better in terms of makespan after testing it with CloudSim Simulator. As we increased a number of virtual machines which are used for simulation, still the total length of scheduling for the proposed algorithm is less than the Genetic Algorithm and FCFS Algorithm.

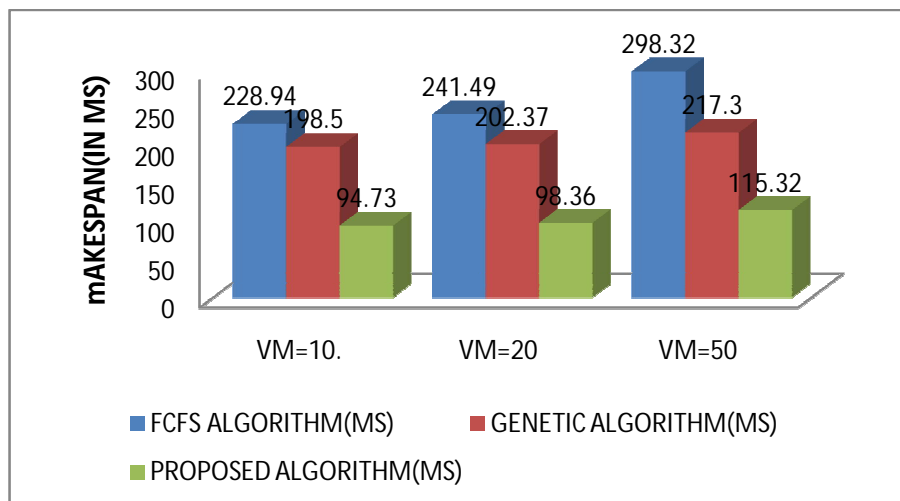


Fig. 3 Comparisons of Three Algorithms.

V. CONCLUSION

In future hybrid task scheduling algorithm can also be developed. Work can be extended for heterogeneous environment also. Parallel and distributed task scheduling methods based on proposed work can also be developed. In future work the effect of precedence between tasks and load balancing will be considered. These methods can also be modeled according to cloud pricing models for providing more benefit to cloud service providers. Separate interface for task scheduling may be provided for cloud controller. The existing scheduling algorithm considered as topic of research and can be used to introduce more efficient and improved performance of algorithms based on parameters like trust value, execution rate, cost of the communication, speed and success rate.

REFERENCES

- [1]. Peter Mell, Timothy Grance, "The NIST definition of Cloud Computing (September, 2011)", Accessed on May, 2014.
- [2]. Bo Li, Tianyu Wo, Chunming Hu, Jinpeng Huai, Lu Liu, and K. P. Lam., "CyberGuarder: A virtualization security assurance architecture for green cloud computing", Future Generation Computer Systems, Vol. 28, Issue 2, pp.379-390, 2012.
- [3]. Rizvi, Syed, Abdul Razaque, and Katie Cover. "Third-Party Auditor (TPA): A Potential Solution for Securing a Cloud Environment." Cyber Security and Cloud Computing (CSCloud), 2015 IEEE 2nd International Conference on. IEEE, 2015
- [4]. Mei, L., Chan, W.K., Tse, and T.H., —A Tale of Clouds: Paradigm Comparisons and Some Thoughts on Research Issues!, In: APSCC 2008, pp. 464-469 (2008).
- [5]. Singh, S., Chana, I.: Q-aware: quality of service based cloud resource provisioning. Comput. Electr. Eng. Elsevier. doi:10.1016/j.compeleceng.2015.02.003
- [6]. Singh, S., Chana, I.: QRSF: QoS-aware resource scheduling framework in cloud computing. J. Supercomput. 71(1), 241–292 (2015).
- [7]. Chana, I., Singh, S.: Quality of service and service level agreements for cloud environments: issues and challenges. In: Cloud Computing- Challenges, Limitations and R&D Solutions, pp. 51–72. Springer International Publishing (2014).



ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 7, July 2017

- [8]. WANG Bei, LI Jun," Load Balancing Task Scheduling based on Multi-Population Genetic Algorithm in Cloud Computing", 35th Chinese Control Conference July 27-29, 2016, Chengdu, China.
- [9]. Salot, P. Et. al. "A survey of various scheduling algorithm in cloud computing environment", International Journal of research and engineering Technology (IJRET), ISSN pp. 2319-1163, 2013.
- [10]. Neeta Patil , Deepak Aeloor," A review – different scheduling algorithms in cloud computing Environment", IEEE 2017.
- [11]. Mayank Mishra, Anwasha Das, Purushottam Kulkarni, and Anirudha Sahoo, "Dynamic Resource Management Using Virtual Machine Migrations", Sep 2012, 0163-6804/12, IEEE Communications Magazine, page no: 34-40.