Task Scheduling in the Cloud Using Machine Learning Classification

Abhijeet P. Tikar
P.G.Student, Dept. of I.T.
Sinhgad college of Engineering,
Pune, India
Email-abhijeettikar@Yahoo.com

S.M.Jaybhaye
Assistant Professor, Dept. of I.T.
Sinhgad college of Engineering,
Pune, India
Email-smjaybhaye.scoe@sinhgad.edu

G.R.Pathak
Associate Professor, Dept. of I.T.
Sinhgad college of Engineering,
Pune, India
Email-grpathak@gmail.com

***Abstract*— Cloud computing is a distributed computing model which enables developers to automatically deploy their applications onto the cloud. There are many applications running on a cloud which requires parallel processing capabilities. Applications of such nature require an efficient scheduling algorithm to manage heavy traffic. The drawbacks of existing scheduling algorithms are low resource utilization and more response time. The aim of the proposed system is to improve resource utilization & response time in the cloud using machine learning classification. Rather than implementing single scheduling algorithms, multiple scheduling algorithms are implemented. Selection of the efficient scheduling algorithm is done using machine learning classification. Initially attributes of the tasks and the Virtual Machine's (VM's) are extracted and used as training data. Training data are given as input to the machine learning algorithm which then produces classification rules. Based on classification rules efficient scheduling algorithm is selected and tasks are executed. The proposed scheme is implemented and tested in the CloudSim simulation toolkit. WEKA tool is used for testing datasets and for the selection of the classification algorithm.**

Keywords- Cloud computing; Task scheduling; Parallel Workloads; Resource utilization.

I. Introduction

Cloud computing is an extension of parallel, grid and distributed computing. Cloud computing provides very secure, fast, user-friendly data storage and processing power with the help of internet. Cloud contains a number of data centers and each data center contains a number of resources. For the data centers an important issue is resource utilization and from user point of view is response time. Scheduling is assigning an appropriate number of resources to the jobs so as to handle heavy load. Main parameters to be considered for scheduling in the cloud are resource utilization, response time, fault tolerance and throughput. Figure 1 shows the basic structure of cloud computing.[2] [3]

Cloud contains a number of data centers. Each data center contains one or more scalable virtual Machines (VM's). Initially user sends input request to the broker then broker is responsible for allocating virtual machine(s) to the user task depending on respective scheduling policy. Finally a task is executed and the result is returned back to the user. [3] [4][14]

The basic algorithm for scheduling in the cloud is First Come First Serve (FCFS). The disadvantage of FCFS is more response time. [5]

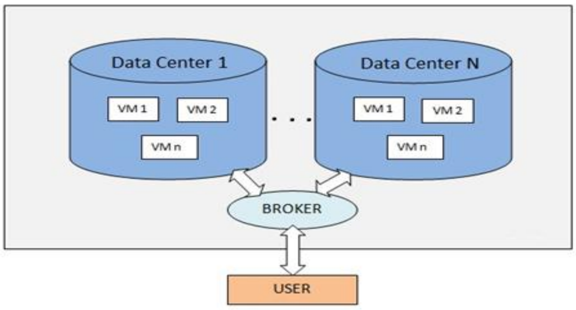


Figure 1. Basic structure of cloud computing

Conservative and Aggressive backfilling algorithms are proposed after FCFS to improve resource utilization and response time. Backfilling algorithms allow shorter jobs to be executed in front of head of queue job if it is waiting for resources. But both these algorithms requires job to mention its maximum execution time. [6] [7] [11]

In this paper, we focus on improving resource utilization & response time on the cloud, as well as system load is also taken into consideration. Each scheduling algorithm in a cloud computing system has some advantages as well as disadvantages. The performance of each scheduling algorithm depends on the type of environment and the type of task. So instead of one algorithm, we have implemented six scheduling algorithms. Selection of efficient scheduling algorithm for the particular cloud environment and input task is done using machine learning classification.

The remainder of this paper is organized as follows: Section 2 discusses some related work. Section 3 presents our task scheduling in cloud using machine learning classification. Section 4 discusses about simulation and results. Section 5 concludes the paper and discusses future work.

II. RELATED WORK

The default algorithm for task scheduling is FCFS. The disadvantage of FCFS is more response time, lower resource utilization and it is non preemptive. [5] Round robin is the simplest algorithm that uses the concept of time slices. In this, the time is divided into multiple slices and each node is given a particular time interval and in this time interval, the node will perform its operations. The disadvantage of this algorithm is the higher power consumption because many nodes kept turning on for a very long time. [12]

Ahuva et al. [6] proposed Extensible Argonne Scheduling system (EASY) which is aggressive version of conservative backfilling method. All the jobs can backfilled in EASY but only the job which at the head of queue can preempt other jobs. Backfilling is allowed if it does not delay the job at the head of queue. Jobs other than the head of the queue may be delayed. The main drawback of this algorithm is that prediction cannot be made about how much each job will be delayed in the queue.

Gang scheduling is proposed to schedule related processes or threads to run simultaneously on different processors. But, Processors remain idle when the system performs I/O operation. To overcome this drawback paired gang scheduling is proposed. [8] [9] [11] Paired gang scheduling matches compute bound jobs with I/O bound jobs and schedule them together. But the disadvantage is prediction should always be accurate while pairing compute bound jobs with I/O bound jobs.[10] [11]

Advancement to Conservative and EASY backfilling is CMBF and AMBF respectively. In addition to the traditional backfilling algorithm, Migration is added so as to improve resource utilization. [1] The disadvantages of CMBF & AMBF are overcome in CMCBF & AMCBF algorithm which divides the capacity of a node into two foreground & background. Foreground having highest priority and background having lowest priority. [1]

From the above analysis, it is concluded that each scheduling algorithm is having its own merits & demerits. Proposed system tries to predict which scheduling algorithm is efficient for the task and cloud environment and task is scheduled & executed based on that scheduling policy. Proposed system improves resource utilization and response time in cloud computing system.

III. SYSTEM ARCHITECTURE

Scheduling refers to a procedure which is used to control the order of work to be performed by a system. A good scheduler readjusts its scheduling strategy, according to the changing cloud environment and the type of input job/task. Figure 2 shows the architecture of the proposed system. In this Data Center configuration and VM configuration is done initially. Data Center configuration includes creation of processing elements & hosts as well as setting up Data Center characteristics. Virtual Machine configuration includes setting up values for virtual memory size, image size, processing power (MIPS) and number of processing elements to create. [14] After creating the cloud environment, cloudlets (tasks), broker as well as scheduling algorithms are designed. Cloudlets are nothing but the tasks which user wants to execute. Broker is responsible for allocating Virtual machine to the cloudlets. In the design of scheduling algorithms we have implemented six scheduling algorithm which includes FCFS, NTS (Novel task scheduling in cloud), CMBF, AMBF, CMCBF and AMCBF.[1] [5] [13] The purpose of VM and attribute extraction block is to extract VM and Cloudlet attributes which are further required for machine learning classification.

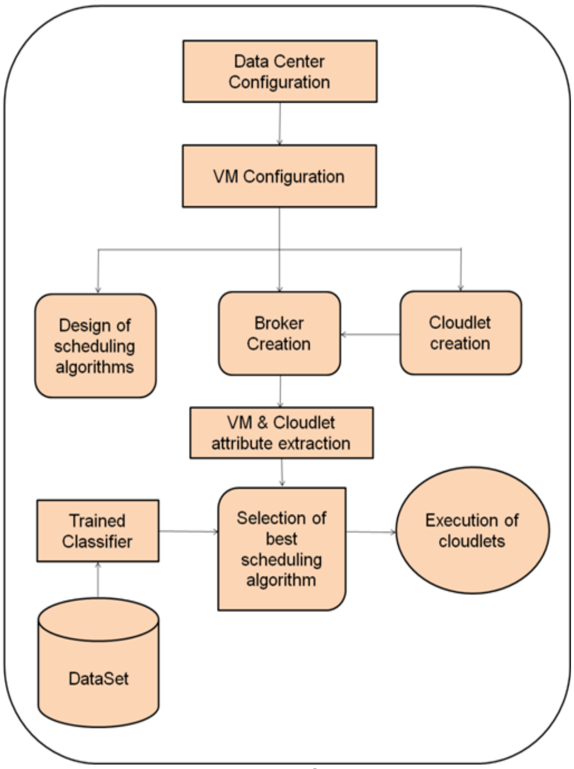


Figure 2. System architecture

The extracted attributes are stored in a text file which we call dataset. Then, Selection of the efficient task scheduling algorithm is done using predicted classification which we will get from trained classifier. Machine learning classifiers are used to train cloudlet and VM attributes. Finally, an efficient task scheduling algorithm will be selected & tasks are executed. [16]

A. *Design of scheduling algorithms*

In the design of scheduling algorithm, we have implemented six scheduling algorithms. Using trained classifier the best algorithm suited for particular cloudlet and VM attributes is chosen and task is scheduled depending on the selected task scheduling algorithm. Scheduling algorithms are as follows:

1. FCFS (First Come First Serve)

The simple & popular batch scheduling algorithm for parallel jobs is first come, first serve (FCFS). In FCFS, each job has to specify the number of nodes required. Processing of jobs by the scheduler is done on the basis of arrival time.[5]

2. Novel approach for task scheduling in the cloud

This is very simple but effective approach where user tasks are sorted according to priority. The task with highest priority is assigned to a VM with highest MIPS (Million Instruction Per Second). The important

parameter for the task is the priority and for virtual machine is processing power. Let's consider 4 tasks where the priorities of tasks are given as 1, 2, 3 and 4. Also consider 4 VM's identified by ID's and each VM having different MIPS as
VM = {{1, 1000}, {2, 100}, {3, 500}, {4, 700}}.
Then task 1 will be assigned to VM1 because, it is having highest MIPS. The remaining tasks 2, 3, 4 will be assigned to VM4, VM3 and VM2 respectively. [13]

3. CMBF -Conservative migration supported backfilling
CMBF is advancement to a traditional backfilling algorithm where migration of the tasks is supported. The CMBF algorithm considers that the state of a job can be saved and can be later resumed on another node. So, the scheduler is able to suspend a job & resume it on another node. If there are enough number of nodes available, then CMBF schedules jobs according to arrival time. [1]

4. AMBF-Aggressive Migration Supported Back Filling.
AMBF is advancement to EASY or aggressive backfilling and alternative to CMBF algorithm. AMBF allows only head of queue job to preempt other jobs. Meaning that rest of jobs are not allowed to preempt jobs, but they are allowed to dispatch to idle nodes. Also AMBF only keeps track of job which is at the head of the queue which saves additional overhead and cost. [1]

5. Priority based consolidation method
Priority based consolidation method divides the computing capacity of a node into two foreground and background. Foreground having the highest priority and background having lowest priority. Priority based consolidation method ensures that background workload does not affect foreground workload. Threshold value is given for every virtual machine; jobs are allowed to run in the background only when the foreground VM has utilization lower than given threshold value. The priority based consolidation method includes two algorithms CMCBF & AMCBF.CMCBF & AMCBF are the advancements to CMBF & AMBF respectively. The difference between CMCBF & AMCBF is in CMCBF any job can backfill and preempts other jobs, but in AMCBF only job at the head of the queue is allowed to preempt other jobs.[1]

B. Selection of efficient task scheduling algorithm using trained classifier

Selection of best scheduling algorithm is done using trained classifiers. In proposed system we have trained cloudlet and VM attributes using machine learning classification. Classification is a technique that involves three stages, a learning stage, a testing stage and an application stage. Classifier is developed during the learning stage. In the testing stage, test data are used to assess the accuracy of classifier. Application stage is used for the classification of new, unclassified data tuples.

Figure 3 explains how to train cloudlet and VM attributes.

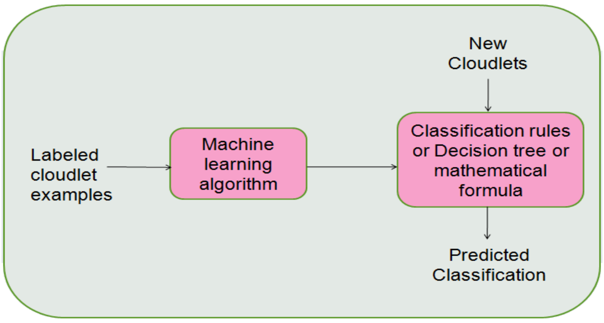


Figure 3 Training Cloudlet and VM attributes

In this, Cloudlets and VM attributes are given as input to machine learning algorithm. Then, machine learning algorithm produces classification rules or decision tree or mathematical formula depending on the type of classification algorithm used. The types of classification algorithm are tree based, rule based, bayes and lazy classification. Tree based classifiers and rule based classifiers produces decision tress and classification rules respectively. This is called as learning phase. In the final application phase, Classification is predicted for new cloudlets and VM attributes.

The classification of decision tree begins as a root node containing all of the training tuples. If all the training tuples are from the same class then the node becomes a leaf. Otherwise tuples are separated into individual classes using splitting criteria. A branch is grown according to splitting criteria and tuples are partitioned. The algorithm recurses to build decision tree for the tuples at each partition.[16] [17].

IV. SIMULATION AND RESULTS

There are many simulators developed for the performance analysis of cloud computing environment, including SPECI, DCSim, CloudSim and GroudSim. But the CloudSim simulator is probably the most sophisticate out of the above mentioned simulators because it is open source toolkit. Also CloudSim supports modeling of cloud computing system components like data centers, host, virtual machines, Cloudlets.[14] [15]

A. Simulation Description

CloudSim version 3.0 is used to implement the proposed system. WEKA tool is also used for selecting machine learning classifier. The simulation is performed on a computer system running Window 7 operating system. The configuration of a computer is as Processor-Intel® Core™ i3, RAM- 4GB , HDD – 1 TB 5400RPM Hard Drive. Following is the basic working procedure of CloudSim Simulator:
1] Initialize CloudSim package
2] Create DataCenter.
3] Create Broker.
4] Create virtual machine.
5] Create Cloudlets.
6] Start Simulation.
7] Stop Simulation.
8] Output the result. [14] [15]

B. DataCenter Creation

DataCenter contains a set of hosts and a host contains set of processing elements. Host is responsible for managing virtual machine during their life cycle. Following are the six steps to create power DataCenter:

1) Create a list to store host machine.
2) Create processing element (PE) list.
3) Create processing elements and add them to the PE list.
4) Create host with its ID, Add list of PE's & Host to the list of host machines.
5) Create DataCenter characteristic object that stores the properties of the DataCenter.
6) Finally create power data center object.

C. Virtual Machine and Cloudlet Configuration

A virtual machine is a software implementation of a machine that runs programs like a physical machine. Cloudlet is a representation of tasks/jobs in CloudSim .The configuration of Virtual Machines & Cloudlet used in this experiment is as shown in Table 1.

TABLE 1.VM & Cloudlet Configuration

Virtual Machine(VM)		**Cloudlets**	
Processing Power	500	No. of cloudlets	10
Image Size	512	Output size	100
RAM	1000	File size	100
No. of processing elements	2	No. of processing elements	2
Scheduling policy	Timeshared	Length	100

D. Execution of tasks in CloudSim

The very first step before starting the simulation in CloudSim is to initialize the CloudSim package and create DataCenter(s) and Virtual Machine(s). Figure 4 shows that numbers of Datacenters entered for creation are 5 and Virtual machine entered for creation are also 5.

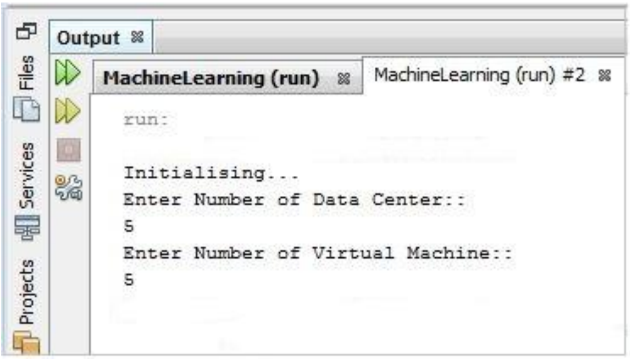


Figure 4. Creation of DataCenter & VM

Figure 5 shows that Datacenters & VM's are created successfully. Also 10 tasks are created from Task 0 to Task 9. The printed format of task is Task=length of cloudlet. Initially at the time of configuration the length of cloudlet was 100. Now length of cloudlet=length*(cloudlet id+1).After creation of tasks, task are submitted to cloudlet list for further processing.

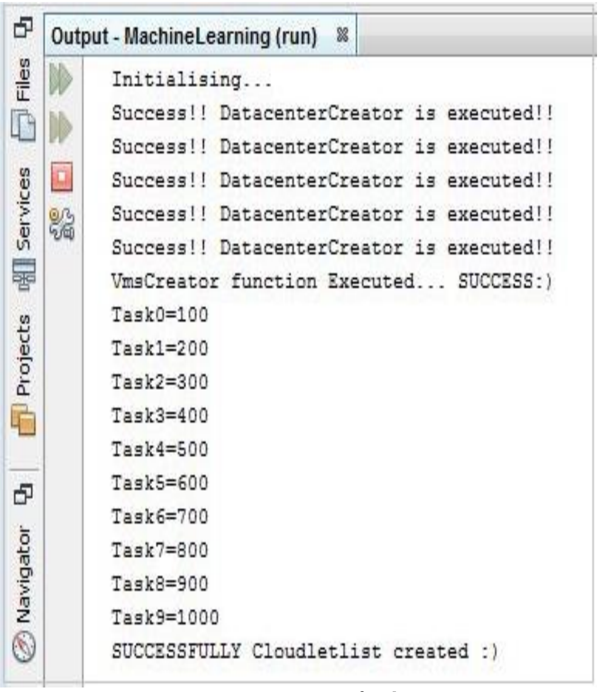


Figure 5. Creation of tasks

Scheduling of above created 10 tasks on to the cloud environment using FCFS broker is shown in figure 6. FCFS broker requests cloudlets from cloudlet list and virtual machine from virtual machine list. Finally, FCFS broker bounds each task with virtual machine in sequential order.

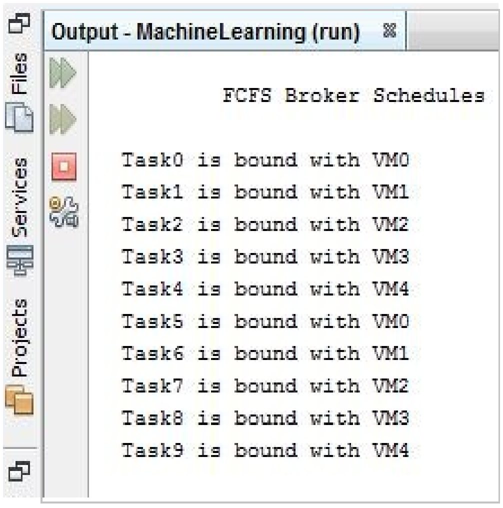


Figure 6. Bounding of Tasks & VM's using FCFS broker

The task which is received first will be executed first. As we can see in figure 7 task (Cloudlet) 1 is received first then 2,5,3,6,4,7,8 and finally 9 is received. After the execution of the entire cloudlets or tasks broker destroys all the virtual machine and then VM broker will shut down. Finally the entire Datacenters & DataCenter broker will shut down.

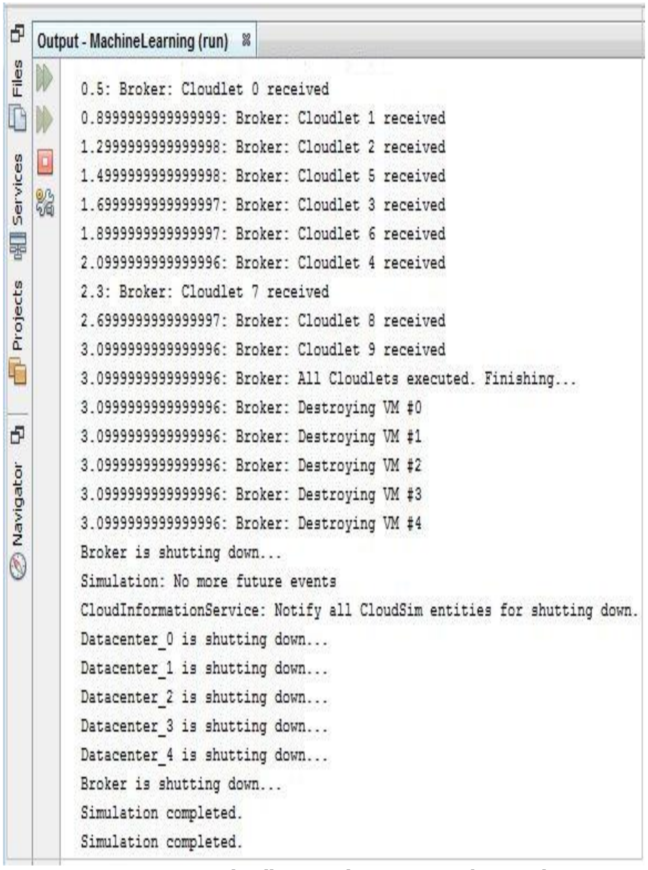
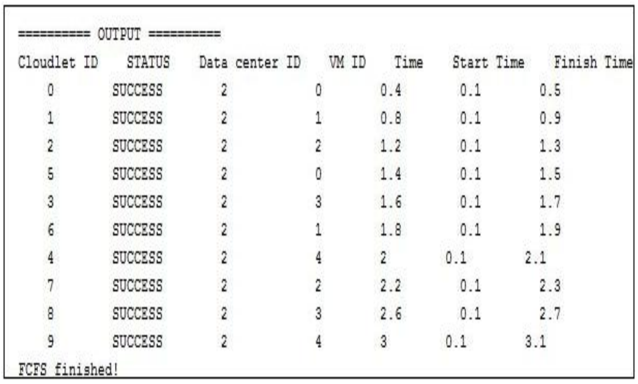


Figure 7. Receiving cloudlet, VM destruction & shutting down Datacenters.

Results of execution of all 10 tasks are shown in figure 8. Output is shown as cloudlet ID, Status, Data Center ID, VM ID, Execution time, Start time and Finish time. For example Status of cloudlet id 0 is success means it is successfully executed in data center ID 2 and VM ID 0.



========== OUTPUT ==========

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
0	SUCCESS	2	0	0.4	0.1	0.5
1	SUCCESS	2	1	0.8	0.1	0.9
2	SUCCESS	2	2	1.2	0.1	1.3
5	SUCCESS	2	0	1.4	0.1	1.5
3	SUCCESS	2	3	1.6	0.1	1.7
6	SUCCESS	2	1	1.8	0.1	1.9
4	SUCCESS	2	4	2	0.1	2.1
7	SUCCESS	2	2	2.2	0.1	2.3
8	SUCCESS	2	3	2.6	0.1	2.7
9	SUCCESS	2	4	3	0.1	3.1

FCFS finished!

Figure 8. Results of execution of tasks

CPU time, start time and finish time of cloudlet ID 0 is 0.4, 0.1 and 0.5 respectively. getActualCPUTime(), cloudlet.getExecStartTime(), cloudlet.getFinishTime() are the functions provided by CloudSim to get CPU time, start time and finish time respectively. In the similar fashion we have implemented 6 scheduling algorithms.

V. CONCLUSION AND FUTURE SCOPE

Proposed system improves resource utilization and response time in cloud computing. The Combination of cloud computing & machine learning classifiers opens a new area for research in cloud. Cloud service providers can use proposed system in their services to maximize profit as well as to use resources efficiently. In the future, we will try to implement more discrete task scheduling algorithms and will try to minimize attributes used in dataset for classification.

ACKNOWLEDGMENT

I hereby take this opportunity to express my heartfelt gratitude towards the people whose help was very useful for the completion of my research work on the topic of "Task scheduling in the cloud using machine learning classification" It is my privilege to express sincerest regards to the project Guide Mrs. S. M. Jaybhaye, for her valuable inputs, able guidance, encouragement, whole-hearted cooperation and constructive criticism throughout the duration of my project work. I deeply express my sincere thank to my Co-guide Mr. G. R. Pathak for his encouragement and support.

REFERENCES

[1] Xiaocheng Liu; Chen Wang; Bing Bing Zhou; Junliang Chen; Ting Yang; Zomaya, AY., "Priority-Based Consolidation of Parallel Workloads in the Cloud," Parallel and Distributed Systems, IEEE Transactions on , vol.24, no.9, pp.1874,1883, Sept. 2013

[2] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. 2010. A view of cloud computing. Commun. ACM *53*, 4 (April 2010), 50-58

[3] Bento, Al, 2011, "Cloud Computing: A New Phase in Information Technology Management", Journal of Information Technology Management, Vol. 22, No. 1, pp. 39-46.

[4] Rao, S, Rao, N &Kumari, K 2009, "Cloud Computing: An Overview", Journal OF Theoretical And Applied Information Technology, Vol 9, No. 1, pp. 71-76.

[5] U. Schwiegelshohn and R. Yahyapour, "Analysis of First-Come-First-Serve Parallel Job Scheduling," Proc. Ninth Ann. ACM-SIAM Symp. Discrete Algorithms, pp. 629-638, 1998.

[6] Mu'alem, Ahuva W., and Dror G. Feitelson. "Utilization, predictability, workloads, and user runtime estimates in scheduling the IBM SP2 with backfilling." Parallel and Distributed Systems, IEEE Transactions on 12.6 (2001): 529-543

[7] Jackson, David, Quinn Snell, and Mark Clement. "Core algorithms of the Maui scheduler." In Job Scheduling Strategies for Parallel Processing, pp. 87-102. Springer Berlin Heidelberg, 2001.

[8] D. Feitelson and M. Jettee,"Improved Utilization and Responsiveness with Gang Scheduling," Proc. Workshop Job Scheduling Strategies for Parallel Processing, pp. 238-261, 1997.

[9] Jonathan Weinberg, "job scheduling on parallel systems", University of California, San Diego, 2001.

[10] Y. Wiseman and D. Feitelson, "Paired Gang Scheduling," IEEE Trans. Parallel and Distributed Systems, vol. 14, no. 6, pp. 581-592, June 2003.

[11] Zhang, Yanyong, Hubertus Franke, Jose Moreira, and Anand Sivasubramaniam. "An integrated approach to parallel scheduling using gang-scheduling, backfilling, and migration." Parallel and Distributed Systems, IEEE Transactions on 14, no. 3 (2003): 236-247.

[12] Vignesh V, Sendhil Kumar KS, Jaisankar N "Resource Management and Scheduling in Cloud Environment"International Journal of Scientific and Research Publications, Volume 3, Issue 6, June 2013 1 ISSN 2250-3153,PP:1-4

[13] Vijayalakshmi, R.; Prathibha, S., "A novel approach for task scheduling in cloud," Computing, Communications and Networking Technologies (ICCCNT),2013 Fourth International Conference on , vol., no., pp.1,5, 4-6 July 2013

[14] Calheiros, Rodrigo N., et al. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." Software: Practice and Experience 41.1 (2011): 23-50.

[15] Buyya, Rajkumar, Rajiv Ranjan, and Rodrigo N. Calheiros. "Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities." High Performance Computing & Simulation, 2009. HPCS'09. International Conference on. IEEE, 2009.

[16] Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques." (2007): 3-24.

[17] Hornik, Kurt, Christian Buchta, and Achim Zeileis. "Open-source machine learning: R meets Weka." Computational Statistics 24, no. 2 (2009): 225-232.