



## Efficient Resource Scheduling in Cloud Computing

Harshil Mehta, Vivek Kumar Prasad and Dr. Madhuri Bhavsar

Computer Science and Engineering  
Institute of Technology  
Nirma University  
Ahmedabad, Gujrat, 382481, India

**Abstract:** Cloud Computing offers elastic, scalable, resource sharing services by using resource management. Resource monitoring and prediction are the keys to achieve resource utilization with high-performance management in cloud computing. Resource scheduling is one of the major issue of cloud computing, the scheduling policy and algorithm affect the performance of cloud system directly. In recent years, Cloud Computing offers high-performance computing capacity, which reminds cloud providers to utilize resource fully because of the limitation of resources. This research paper aims to monitor the resources available in cloud using Hidden Markov Model (HMM). The proposed model is used for resource monitoring and then the resource will be classified based on Less, Average, and Heavy loaded categories as the availability of the resources and the appropriate scheduling algorithm will be selected on demand, the efficiency of algorithm has been calibrated using different kind of workload scenario.

**Keywords:** Cloud Computing, High Performance, Resource Classifier, Utilization, Hidden Markov Model (HMM)

### I. INTRODUCTION

Many of the research and development industries expressed their ideas and interest in the field of cloud computing. Cloud computing mainly refers to provide on-demand services with the better utilization of the resources which is used by the each and every host in the cluster. It also provides the flexible manner of the provisioning, of the resource for giving faster and efficient result to the user. Resource scheduling and allocation are mainly considered to be a critical issue in Cloud computing. Generally, it is difficult or hard to find the optimal resource allocation which minimizes the execution-speed and effectively utilized the resource to which is available.

#### A. Services Provided By Cloud

**Software as a Service** To complete the Business tasks Business user uses the SaaS Model to delivered the web application. SaaS provides Automation, Customer Relationship Model, Website testing and many more services

**Platform as a Service** Developers and Deplorers uses the PaaS model to create or deploy application and services for users. Application test, development, integration and deployment, kind of services provided by the PaaS model

**Infrastructure as a Service** System manager uses the IaaS model to create platform for service and application testing. Virtual machine, operating system message queue, network, storage, CPU, memory, backup services are available in IaaS Model.

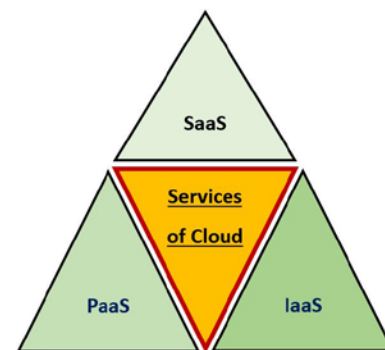


Fig. 1: Services Provided By Cloud

Service of Cloud	Users	Available Services	Use
IaaS	System Manager	Virtual Machines, OS, Message Queues, Storage, CPU, Memory, Services	Create platform for services and applications test, Development integration and Deployment
PaaS	Developers and Deployers	Services and Application test, development integration and deployment	Create or deploy application and services for users
SaaS	Business Users	Emails, Office automation, CRM, Website testing, Wiki, Blog	To complete desktop task

Fig. 2: Services Provided By Cloud [1]

Cloud computing mainly refers to provide on-demand services with the better utilization of the resources which is used by the each and every host in the datacenter. It also provide the flexible manner to provisioning of the resource for giving faster and efficient result to the user. Resource scheduling and allocation is mainly considered to be a critical issue in Cloud computing. Generally it is difficult or hard to find and optimal resource

allocation which minimize the execution and effectively utilized the resource which is available.

### B. Resource monitoring and tools

Commercial Platform	Open sourcePlatform	Services
CloudWatch	Nagios	CloudSleuth
AzureWatch	OpenNebula	CloudHarmony
CloudKick	CloudStack ZenPack	Cloudstone
CloudStatus	Nimbus	Cloud CMP
Nimsoft	PCMONS	CloudClimate
Monitis	DARGOS	Cloudyn
LogicMonitor	Hyperic-HQ	Up.time
Aneka	Sensu	Cloudfloor
GroundWork		CloudCruiser
		Boundary
		New Relic

Fig. 3: Resource Monitoring tools [2]

Cloud resource monitoring is a process to provide efficient cost and output to the customer, it is also beneficial for the cloud providers to control and maintain the software and infrastructure.

### C. Task scheduling

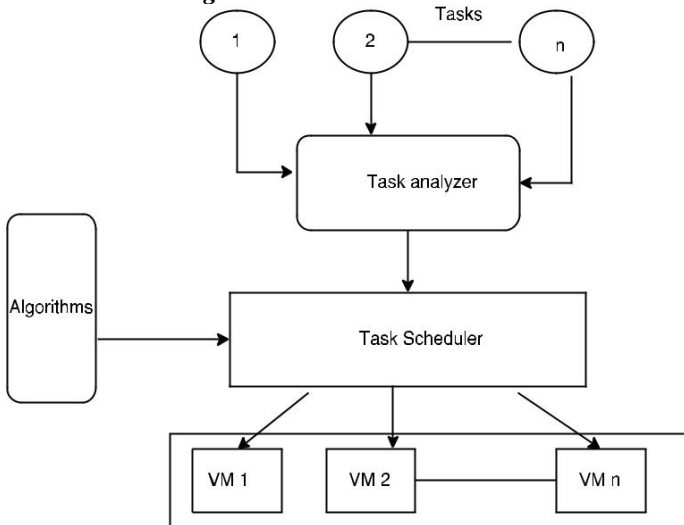


Fig. 4: task scheduling process

There are multiple tasks arriving at the task scheduler as a request type. First task analyzer checks, which kind of task have been arrived as request, for example, there are many types of tasks like dependent and independent, real-time and non real-time etc. Then it is scheduled based on their priority and requirement

A Large number of scheduling algorithm is available to effectively utilized the resource and minimize their computational cost. Resource scheduling is the key process for the infrastructure clouds. IaaS cloud generally takes virtual machine as a scheduling nodes. There are some popular open-source cloud systems, such as Eucalyptus, Open Nebula, and

Nimbus, to decide the allocation of resources. To decide the resource scheduling and allocation Eucalyptus uses the greedy method with allocating the resource based on first fit method. And it also uses the round robin method. Where open nebula uses the queuing technique or advances reservation method for the scheduling of the resource. These all use the random approaches for the utilization of the resource in some cases they do not considering the maximum usage of the resource.

In cloud computing, load balancing [3], [4] is another important technology which comes under the cloud computing platform. As the number of user increases for the cloud resources the cloud resources should have the capability to servicing to the n number of users request. For the better availability of cloud resources and for the shorter response time for the user request, [14] load balancing plays a main role in IaaS cloud computing environment.

## II. RELATED WORK

Authors have proposed many monitoring tools [2] which are providing monitoring of resources. While monitoring of resources is beneficial for the CSP as well as consumers. There are many monitoring tools described in this paper, which some of are commercial and some of are open source. with the help of this monitoring tools we can monitor the resources like CPU, RAM, Memory, Throughput. Managing and controlling of the resources like hardware and software is key performance for future application.

Mell, Peter and Grance, Tim [5] have presented in this paper, that Now a days monitoring of the Cloud resource is important to the both the users and service provider, CSP have to provide the services to the user at a time that means on Demand, the provided Resource is under utilized or over utilized the have to be measured for both the perspective. More analyzing and monitoring tools which measure the performance of the cloud and it also gives the analytical result.

Luo, Liang and Wu, Wenjun and Di, Dichen and Zhang, Fei and Yan, Yizhou and Mao, Yaokuan [6] have presented an algorithm for the cloud computing environment. In this proposed solution they allocates the resource based on energy optimization scheme or based on preallocation of the resource. In this research they have tried to meet the quality of service through preallocation the resource to the VM. Limitation of this proposed method is Resource preallocate in static manner that's why it doesn't knows the future task will take how much time or resource from current Physical Machine.

Zhong, Hai and Tao, Kun and Zhang, Xuejie [7] have used Improved Genetic Algorithm instead of genetic algorithm. They ensure that this policy is much better then the Tradition Genetic Algorithm policy. While they perform the Improved genetic algorithm they measure that IGA have twice a speed then TGA this is not consider the maximum usage or not considering the utilization rate thats the limitation of this method.

Chen, Hanxiong and Fu, Xiong and Tang, Zhongrui and Zhu, Xinxin [8] have proposed a method for monitor and prediction problem of present scenario of cloud culture. They have used vector auto regression by correlation between resource and trying to predict and monitor the present status. Proposed model performs better but they conclude that time complexity for prediction for resources is more then they try to consume.

Shin, SaeMi and Kim, Yena and Lee, SuKyoung [9] have proposed conservative backfill algorithm which uses the earliest deadline first or largest weight first algorithm in this method when job arrives at datacenter it sort the job as in priority and schedule them. If the dependent task may come then this solution doesn't work, which is its limitation.

Han, Yaojun and Luo, Xuemei [10] author have proposed Least Language First MinMin algorithm which is based on exist min min algorithm. Which of the task have minimum number of site which selects first to execute or list of task have created. This method has poor load balancing and some QoS factor is not being considered which is its limitation.

Vivek Kumar Prasad [11] have proposed the load balancing technique and scheduling of tasks in parallel processing environment. Author have used HMM to predict load of the node into the network and tried to assign correct task to the correct processes in the distributed system.

### III PROPOSED ALGORITHM

---

#### Algorithm 1 Resource Efficient Classifier

---

```

1: Initialization
2: Input: Task ti from task queue Ti 1, 2, 3n
3: Output: Classification of Resource based on their load
4: T[i] = Number of task
5: Th = Throughput
6: Wsi = Weight of Server
7: Wsi=n= Weight of all Server
8: HMM = Hidden Markow Model

9: {
10: Get from user(job_info)
11: Get from Resource(resource_utilization)
12: Specified task Resource Requirement
13: /*Find the resource utilization percentage*/
14: Check resource availability
15: Calculate CPU availability=Total available CPU Needed By task
    T[i]
16: Calculate Memory availability = Total available Memory-
    Needed By task T[i]
17: Calculate Storage availability = Total available Storage Needed
    By task T[i]
18: if Resource Availability >= Task needed then
19:   {
20:   Do prepare list for Execution;
```

```

21:   }
22: end if
23: /*Find the Throughput based on Given resource*/
24: Prepare list with Execution time by task
25: Prepare list with average Execution time of each task
26: Thz = No. of Task * Average Execution time of each task

27: Find usage of the resources in the datacenter
28: Observing the usage pattern using HMM
29: Classified the resource Based on their state
30: Based on certain usage value defining the current state of
    resources

31: State 1 Less utilize
32: State 2 Average utilize
33: State 3 Heavy utilize
34: Apply the scheduling algorithm dynamically to the re-sources

35: if State 1 = usage below 30% then
36:   {
37:   Then Resource is under utilized
38:   Apply appropriate scheduling algorithm from the scheduling
    library

39:   }

40: else
41:   (State 2 = usage above 30% and below 70%)

42:   {

43:   Then resource is average utilized
44:   Apply appropriate scheduling algorithm from the scheduling
    library

45:   }

46: else
47:   (State 3 = usage above 70% )

48:   {

49:   Then resource is heavy utilized
50:   Apply appropriate scheduling algorithm from the scheduling
    library

51:   }

52: end if
53: }
```

---

#### IV. PROPOSED WORKING MODEL

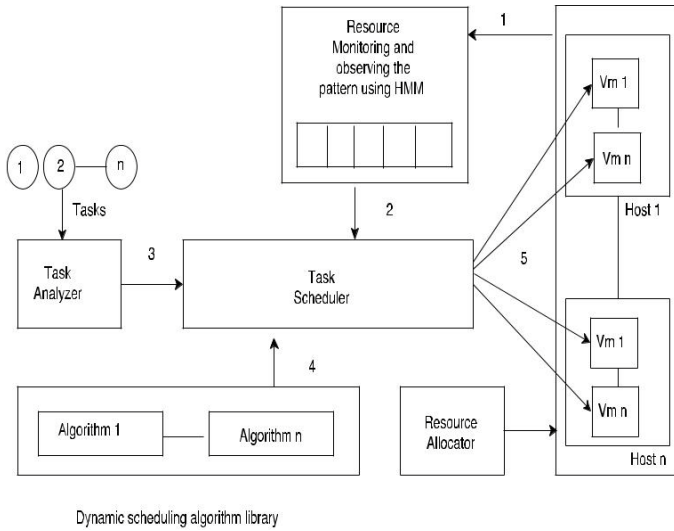


Fig. 5: The flow of proposed working model

##### Step 1:- Resource Monitor

Resource monitor firstly gets the information state of the allocated task to the different VMs and as well as profiles the each VM at given amount of time interval after getting the information of Host and VM, it reports to the task scheduler.

##### Step 2:- Task analyzer and scheduler

Task analyzer gather the information of newly arrived task, it also measures the requirements of the task like arrival time of the task, deadline of the task and execution time of the task, etc. then it schedules the task using task scheduler.

##### Step 3:- Scheduling algorithm

As per the task requirements and current resource usage appropriate scheduling algorithm will be selected dynamically for the particular VM.

##### A. Resource monitoring using HMM

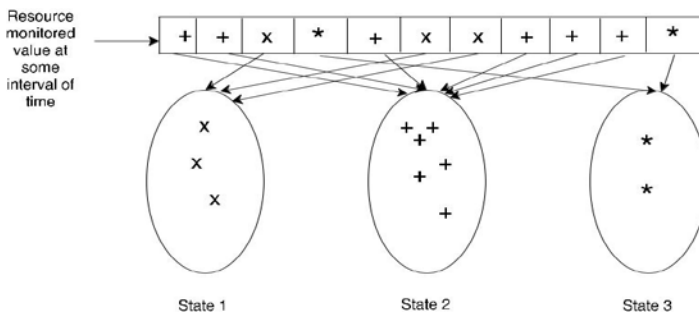


Fig. 6: Observing usage pattern using HMM

For the resource monitoring purpose we have used an open-source tool that monitors host as well as VM. That gives the

CPU and RAM usage values at certain point of time interval. After getting the values, we observe the usage pattern using HMM as shown in figure 6, [12] then we classify the state based on that observed value and will apply the scheduling algorithm dynamically to the particular host into the datacenter [11] [13].

As shown in figure 6 there are defining three states, first state stands for less utilization, second state stands for average utilization, and third for heavy utilization. Into the figure there are defining three symbols "x", "+" and "\*". Symbol x define the value below 30 percent usage of resources, where + symbol define the above 30 and below 70 percent usage of resources, and symbol \* define the above 70 percent usage of resources into the datacenter.

##### Step 4:- Resource allocator

If the task cannot be finished within expected amount of time then resource allocator will create the new VM as per the task requirement and if the some of the resource is lightly loaded or may be idle then it consolidates the VMs.

##### Step 5:- Task allocation

Task allocator gets all the information, then it will be allocating the task to the other VM for the execution.

#### V. EXPERIMENTATION AND RESULT ANALYSIS

For the experimentation we have used two different system in the test for proposed work, one has Ubuntu Operating System and another system has Windows Operating System where are cloud has been installed. This system was monitored using open source cloud tool. Using monitored of the resource at 5 Minute of an interval of time. And for the scheduling purpose, we have used cloudsim and Workflowsim as task simulation.

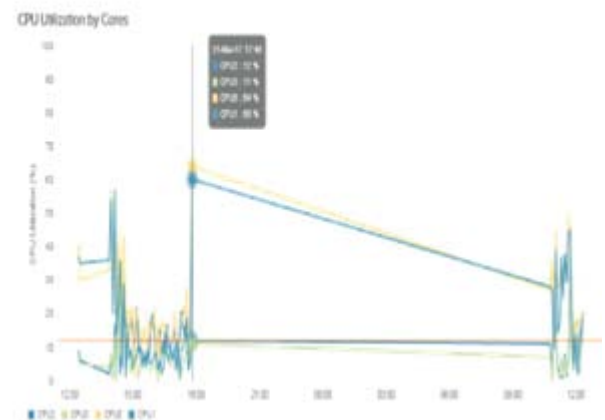


Fig. 7: CPU usage 1

TABLE I: Data observed during Experimental

Date and Time	CPU Usage	RAM Usage
April 9 2017 8:33 AMIST	16.02	54
April 9 2017 8:38 AMIST	40.87	67
April 9 2017 8:43 AMIST	34.38	74
April 9 2017 8:48 AMIST	28.2	69
April 9 2017 8:53 AMIST	13.88	68
April 9 2017 8:59 AMIST	78.99	74
April 9 2017 9:05 AMIST	72.3	71
April 9 2017 9:11 AMIST	83.26	73
April 9 2017 9:16 AMIST	65.3	52
April 9 2017 9:21 AMIST	68.80	44

Graph of the CPU usage is shown in Figure 7 and 8 describes the resource usage rate at some interval of time, figure 7 shows the usage rate of system at the time 6:00 PM in this scenario observed four CPU core, two of them is heavy utilized and other two is less utilized and the other figure describes the usage at 12:00 AM where all CPU cores are average utilized. We observed the 10 results of the CPU and RAM at the 5 minute of interval that has been described below.



Fig. 8: CPU usage 2

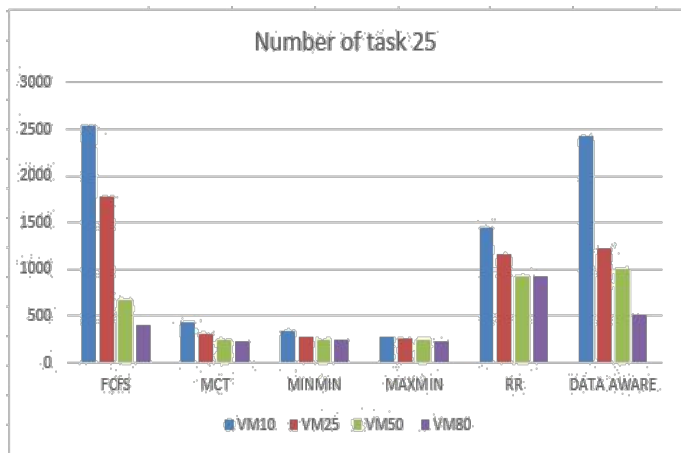


Fig. 9: Analysis

Above table shows the CPU usage and RAM usage rate at the 5 minute of interval time. Our resource monitoring tool that will monitor the on premise poller system and frequently checks the system usage. Based upon the observation we determine that

five reading are in the state 3, two are in state 2 and rest in state 1. Hence using HMM we determine more readings are in state 3, so the system is heavy loaded. Based upon this scenario appropriate scheduling algorithm is invoked using dynamic scheduling mechanism.

TABLE II: Execution time of 25 task

Algorithm	VM 10	VM 25	VM 50	VM 80
FCFS	2541.95	1782.58	677.45	401.01
MCT	430.20	314.6	252.8	236.8
MINMIN	339.2	270.8	240.3	244
MAXMIN	274.3	259.40	248.4	238.2
RR	1449.4	1158.5	930.95	924.57
DATA AWARE	2435.04	1221.65	1009.09	518.25

As in figure 10 shows the FCFS algorithm has the longer execution time, and Max-min has the smallest execution time, where Round-robin has the average execution time.

Above table shows the execution time of the different algorithms for the 25 tasks where the VM has been increased upto 80 and their deviations in the execution time has been described into the above graph.

TABLE III: Execution time of 50 task

Algorithm	VM 10	VM 25	VM 50	VM 80
FCFS	7975.86	6376.37	3813.9	2248.7
MCT	2203.8	947.3	681	681
MINMIN	3424	3367.10	755.67	596.15
MAXMIN	1619	641	687.3	592
RR	3563.7	2386.5	2078.4	687.52
DATA AWARE	7504.91	4168.72	1259.75	1258.46

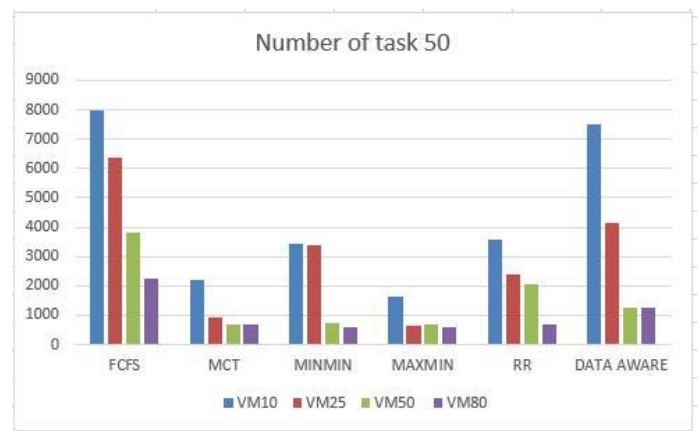


Fig. 10: Analysis



TABLE IV: Execution time of 100 task

Algorithm	VM 10	VM 25	VM 50	VM 80
FCFS	5084.9	3210.1	2675.6	2390.2
MCT	3844	2500	2489	1662
MINMIN	5291	3641	2983	1849
MAXMIN	3036	1986	1745	1013.2
RR	3821.7	2676.6	2762	2543.7
DATA AWARE	5451.12	3638.23	2717.97	2060.86

As above mentioned that, for the another experimental purpose tasks has been increased, above table shows the execution time with respect to different VM which has been increased. Resources in terms of Virtual Machine.

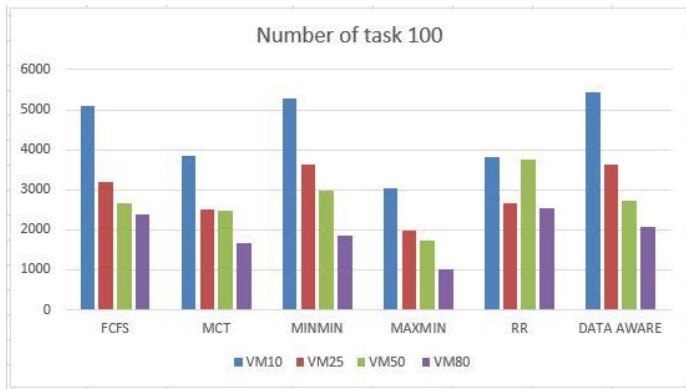


Fig. 11: Analysis

As per the third table that describes the execution time of 100 tasks with respect to various scheduling algorithm, based upon the availability of the resources in cloud computing environment. As per the observation Max-Min algorithm is the efficient and FCFS is less efficient algorithm.

## VI. CONCLUSION

Cloud Computing has the dynamic nature in provisioning of resources in static and dynamically. For the resource monitoring purpose open source tools have been used into the research, and resources were classified using HMM. In the proposed method for better utilization of the resource a classifier is to be developed that classifies utilization in three major categories 1) Less utilization 2) Average Utilization 3) Heavy utilization with using HMM. Based upon this usage pattern different algorithm can be classified for different conditions.

## VII. FUTURE SCOPE

In future focus will be on context-aware scheduling algorithm in cloud computing and other parameters also be considered apart from CPU and RAM.

## VIII. ACKNOWLEDGMENT

It gives me immense pleasure in expressing thanks and profound gratitude to Dr. Madhuri Bhavsar Head of Department and Prof. Vivek K. Prasad, Assistant Professor, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad for them valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

## IX. REFERENCES

- [1] "Image: Nist definition for saas, paas, iaas — cloudinfosec." <https://www.google.co.in/imgres?imgurl=https> Accessed: 2016-09-15.
- [2] G. Aceto, A. Botta, W. De Donato, and A. Pescapé, "Cloud monitoring: A survey," *Computer Networks*, vol. 57, no. 9, pp. 2093–2115, 2013.
- [3] I. Psoroulas, I. Anagnostopoulos, V. Loumos, and E. Kayafas, "A study of the parameters concerning load balancing algorithms," *IJCSNS International Journal of Computer Science and Network Security*, vol. 7, no. 4, pp. 202–214, 2007.
- [4] S. Sharma, S. Singh, and M. Sharma, "Performance analysis of load balancing algorithms," *World Academy of Science, Engineering and Technology*, vol. 38, no. 3, pp. 269–272, 2008.
- [5] P. Mell, T. Grance, et al., "The nist definition of cloud computing," 2011.
- [6] L. Luo, W. Wu, D. Di, F. Zhang, Y. Yan, and Y. Mao, "A resource scheduling algorithm of cloud computing based on energy efficient optimization methods," in *Green Computing Conference (IGCC), 2012 International*, pp. 1–6, IEEE, 2012.
- [7] H. Zhong, K. Tao, and X. Zhang, "An approach to optimized resource scheduling algorithm for open-source cloud systems," in *2010 Fifth Annual China Grid Conference*, pp. 124–129, IEEE, 2010.
- [8] H. Chen, X. Fu, Z. Tang, and X. Zhu, "Resource monitoring and prediction in cloud computing environments," in *Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence (ACIT-CSI), 2015 3rd International Conference on*, pp. 288–292, IEEE, 2015.
- [9] S. Shin, Y. Kim, and S. Lee, "Deadline-guaranteed scheduling algorithm with improved resource utilization for cloud computing," in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 814–819, IEEE, 2015.
- [10] Y. Han and X. Luo, "An effective algorithm and modeling for information resources scheduling in cloud computing," in *Advanced Cloud and Big Data (CBD), 2013 International Conference on*, pp. 14–19, IEEE, 2013.

- [11] V. K. Prasad, "Load balancing and scheduling of tasks in parallel processing environment,"
- [12] R. Patidar, L. Sharma, et al., "Credit card fraud detection using neural network," International Journal of Soft Computing and Engineering (IJSCE), vol. 1, no. 32-38, 2011.
- [13] A. Srivastava, A. Kundu, S. Sural, and A. Majumdar, "Credit card fraud detection using hidden markov model," IEEE Transactions on dependable and secure computing, vol. 5, no. 1, pp. 37–48, 2008.
- [14] A. Revar, M. Andhariya, D. Sutariya, and M. Bhavsar, "Load balancing in grid environment using machine learning-innovative approach, "International Journal of Computer Applications, vol. 8, no. 10, pp. 31–34, 2010.