DYNAMIC MIN-MAX ALGORITHM FOR RESOURCE PROVISIONING IN CLOUD ENVIRONMENT

Chethan Venkatesh

Assistant Professor, Department of MCA,
M.S.Ramaiah Institute of Technology,
Research Scholar, VTU RRC
Chethu4u@gmail.com

Shiva Murthy G

Associate Professor, Department of MCA,
VTU center for PG Studies, Bangalore
kgshivam@gmail.com

Abstract

Cloud computing is a service-oriented architecture. Cloud offers different service models such as Software as a Service(SaaS), Infrastructure as a Service(IaaS) and Platform as a service(PaaS). The core component of every cloud is resource provisioning. Resources such as CPU, memory, and storage are provisioned dynamically to support cloud applications. This paper addresses scheduling of the tasks dynamically by taking into account the total execution or completion time of the tasks and resource utilization. The Cloud scheduler queries the Cloud Information System to check for the availability of resources, knowing their properties, and then scheduling the resources as per task requirements. The results are sent back to users once the task completes. The challenge lies in scheduling the tasks in a heterogeneous environment.

***Keywords*:** Cloud computing; Resource provisioning; Data centers; Dynamic Min-Max

1. Introduction

Cloud computing offers resources such as CPU cycles, storage, memory, bandwidth, virtual machines etc... On a pay-as-you-use model, where a customer can use the resources as per his requirement. These resources are allocated according to the availability of resources and the demand from the customer. Once the task of the customer is completed, the resources are deallocated and the customer is billed on a pro-rata basis. The amount of power and speed in Cloud Computing is realized from virtualization. Virtualization enables hardware resources to be effectively utilized. In order to minimize the usage of physical servers, virtual machines are allocated based on the need for users. That being said, most VM resources are not effectively utilised on the basis of work characteristics to comply with Service Level Agreements (SLA), resulting in low usage of resources.

Cloud computing enables personal users and companies to consume resources such as virtual machines, storage, applications as utility and with a lowest pricing and with pay per usage model helps companies to save huge amount in purchasing and maintaining physical servers.

Several models are proposed for resource provisioning which addresses the optimization issues involved in provisioning so as to minimize several parameters such as cost, response time etc.… One of the popular approaches we have found is the min-max approach where in the incoming requests are assigned the resources for execution. The tasks which arrive are combined in batches by sorting them and then each batch is submitted for execution. New tasks which arrive are accommodated in a new batch. The batch is submitted once it is full and hence tasks which arrive early will have to wait for a long time to get the resources.

In this paper we propose a resource provisioning mechanism which mainly takes care of minimizing the execution time of the tasks and also balances the load among multiple resources dynamically. The proposed work analyses the efficiency of resource provisioning in the cloud computing environment. Research is carried out to provide a sustainable foundation for the existing system. To achieve the objective, a simulation environment is set up to highlight the performance of real-time systems in the cloud datacenter. The objective of the proposed work is to achieve the target of minimizing the execution time of the tasks. The simulation is carried out with varying workloads. It allows scaling up and scaling down the resources according to the job requirements. The need of on-demand resource provisioning or dynamic resource provisioning arises here,

[1][2]where the virtual machines are allocated and deallocated to users as and when the need arises. In order to deploy applications in a highly available and load balanced environment, effective resource utilization mechanisms have to be considered. We propose a dynamic VM allocation algorithm that dynamically reconfigures virtual resources considering the characteristics of the job, which in turn improves the utilization of resources. Over all there is an improvement in the throughput of the system and the resources are being utilized efficiently. We also present the evaluation results of both min-max algorithm and our approach.

Section 2 discusses related work, following the architecture of the proposed system resource provisioning scheme in section 3. The proposed algorithm is discussed in section 4. Section 5 describes the experimental setup, the results and deliberation are presented in section 6 and the conclusion in section7.

2. Related Work

N. Susila et al., [3] proposed an approach for analyzing the performance in a distributed cloud system of heterogeneous nature. A research work is carried out in order to extend a solid foundation that facilitates the significance of the current structure, research work is being carried out. The key goals aimed at accomplishing the objective are to minimize the number of servers used in the datacenter's storage functionality and also to balance the load. In a simulation environment with a Miscellaneous distribution of workloads, an Energy Efficient Load Balancing (EELB) using the First-Come-First-Served (FCFS) approach is presented to examine the functionality. The technique of triggering the VMs to SLEEP mode is mainly involved in the algorithm when it is either in idle mode or when it is heavily used. When underused, i.e. in an idle state, and in an overused state, it is put in a sleep mode. The results of the experimental study confirm energy efficiency in the storage system of the datacenter in tandem with efficient load balancing compared to the current system.

Abdul Hussein, Abdul Mohson et al. (2015) [4] have proposed a genetic algorithm whose main intention is based on live Virtual machine migration. The algorithm is a search heuristic that replicates the process of natural selection. Steps in Genetic algorithm involve Population Coding, Initialization of Population, Fitness Function, Selection Strategy, Crossover and Mutation.

Srinivas Sethi, Anupama Sahu et al., in [5] have proposed a novel round-robin based load balancing algorithm in Virtual Machine (VM) cloud computing environment. It holds the state information in each VM in addition to the number of tasks currently allocated to every VM's, The algorithm identifies the VM that is least loaded, if a new request to allocate arrives, the first least loaded machine (if more than one) is identified. By applying the fuzzy logic, it takes two inputs like processor speed and allocated load of virtual machine and converts them into one output such as balanced load in the inference system.

Zhang Qian, Ge Yufei et al., [6] have proposed a Task scheduling algorithm based on peer-to-peer cloud computing environment. Every node can receive the tasks, evaluate the available resources, schedule and execute the tasks. The algorithm makes use of weighted random strategy, overload assessment and feedback and ensure that efficient nodes are not overloaded when tasks are submitted to the best resource. It also ensures that resources with normal performance continues to execute the tasks. The algorithm will balance the load effectively and balances the workload of the nodes in the network and presents a solution in the cloud computing environment for the load balancing strategy.

Improved Max-Min algorithm [7] is an effective load balancing technique, Load balancing technique is applied to obtain gain better performance and effective resource utilization. In the proposed algorithm, for each resource, it calculates the estimated completion time of all the tasks submitted. The tasks with the highest estimated execution time are then assigned with the resource with the minimal overall completion time. The task is then deleted and the estimated times are changed from the meta-tasks. Until all the tasks are executed, this process is repeated. The algorithm minimizes the total duration of output. In cloud computing environments tasks are executed concurrently on available resources to achieve better load.

Sagar D. Girase, Mayank Sohani et al., [8] have proposed a priority based design and implementation of a resource management system that achieves maximum resource utilization with minimal response time. Cloud storage facilities rely on efficient planning and resource management policies for their effectiveness. In distributed systems, the goal and objective of scheduling technique is to distribute the workload on processors to obtain maximal utilization and minimal total request execution time. The authors have presented a scheduling algorithm which addresses the issue of resource contention on virtualized cloud environments. The system includes a strategy for VM allocation, a front end load balancer and a mechanism for pre-emption. The proposed mechanism helps to pick the request, based on its capacity, from the waiting queue. The results of the simulation show that the proposed mechanism performs better than the current available mechanisms, provided that the resource capacity is high.

3. Proposed Work

The architecture of the proposed work is given below in figure -1. The architecture consists of multiple tasks, a datacenter broker, virtual machine scheduler, and the data center consists of ‘n’ hosts and each host having two virtual machines. The configuration of the datacenter is described in the experimental setup.

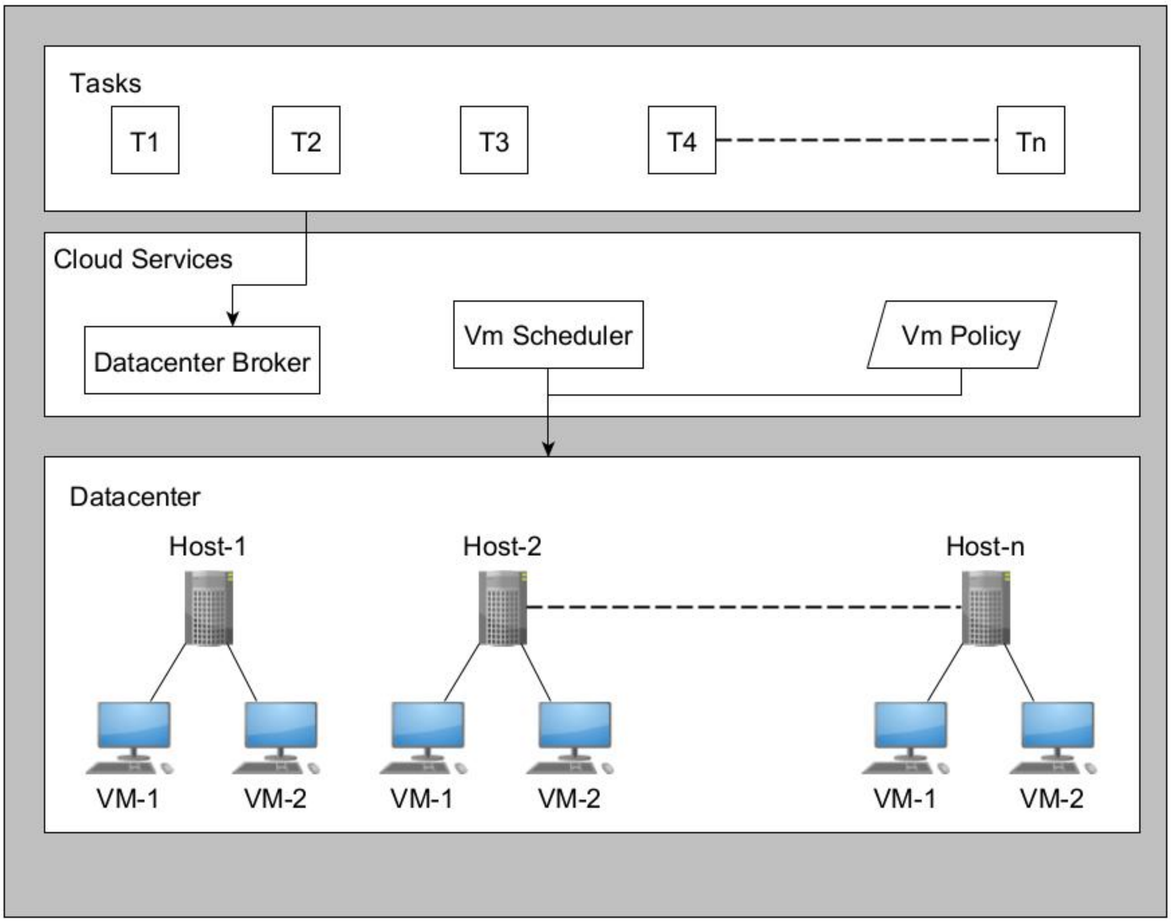

Fig 1: System Architecture of "Dynamic Min-Max"

In Fig 1 the complete system architecture of the “Dynamic Resource Provisioning in Cloud Environment” application is described. The system architecture mainly consists of the Resource provisioning and its sub modules. There are two sub-modules present; one is the CIS and Datacenter, which is responsible for computing Estimation time of cloudlets and assign VM(s) accordingly. The system consists of tasks, CIS and Datacenters. Which intern interacts with each other to complete the execution of task.VMScheduler will schedule each job to specific VM(s) based on VMPolicy.

There are 3 state of cloudlets, before it is finished executing, In the first stage Cloudlets check for available resources, if available then cloudlets are transferred to execution state or if resources are not available then cloudlets have to wait until The resources have finished previously submitted cloudlets, If all the cloudlets have finished execution then cloudlets are submitted to Finished state and records the execution time and VM will move to shutdown state.

4. Algorithm for Dynamic Min-Max Resource Provisioning

Consider M as execution time required for a cloudlet to execute on a VM and $\boldsymbol{L}$ is length of the cloudlet. Here, $\boldsymbol{P}$ is number of processing elements in a VM and $\boldsymbol{mips}$ is the millions of instructions that a VM can process every second. In the algorithm, $\boldsymbol{t_{tat}}$ is the total turnaround time for a cloudlet, $\boldsymbol{t_{rem}}$ is the time remaining for the execution of the cloudlet which is executing on the VM and it will be the waiting time for the next cloudlet allocated to the VM for processing. Also, $\boldsymbol{Ccur}$ is number of cloudlets currently allocated to the VM for processing

```
Step1: Create data centers
Step2: Create Hosts in the datacenter by specifying the Configuration.
Step3: Create VM's by specifying the Configuration.
Step4: Create Cloudlets by Specifying Cloudlet Configuration.
Step5: Create datacenter Broker.
Step6: For all $VM_i$ created and cloudlets $C_j$

If VM has no tasks allocated
then
        Calculate $\boldsymbol{M = (L/(p * mips))/1000}$
        Allocate $Cj$ to $VMi$ with least execution time
Else
        If $((\boldsymbol{t_{tat}} < (\boldsymbol{t_{rem}} + \boldsymbol{t_{ex}}))$
then
        Allocate $Cj$ to $VMi$ with least execution time
```

This algorithm mainly focus on efficient resource allocation to its users for faster execution of tasks we present a VM selection algorithm that seeks to find those VMs with the most efficient in executing tasks , For each request from the user algorithm computes the expected execution time and allocates those tasks to the VM which takes minimal completion time

In a real scenario, not all tasks are ready for execution at a current time, hence this algorithm keeps track on VM about its ready time

5. Experimental Setup

The simulation was carried out on CloudsSim. CloudSim is a tool used to model and simulate cloud architecture. It provides a platform for the researchers to carry out modeling and simulation for large cloud environment. It has built-in nodes, hosts, data centers and packages to support provisioning of resources [9].

We have configured the machines as given below in table 1

VIRTUAL MACHINES	RAM	MIPS
VM1 , VM2	512	250
VM3 , VM4	1024	270
VM5 , VM6	2048	290

Table 1 Virtual Machine Configurations

6. Results and Discussion

Simulation of over 10000 jobs has been carried out and the descriptive statistics are mentioned in table 2

min-max		*D-min-max*	
Mean	370.4066093	Mean	370.4075305
Standard Error	2.153890264	Standard Error	2.153948374
Median	368.93	Median	369.07
Mode	1.92	Mode	32
Standard Deviation	215.3674864	Standard Deviation	215.3732969
Sample Variance	46383.15422	Sample Variance	46385.65701
Kurtosis	-1.146638267	Kurtosis	-1.146544565
Skewness	0.037814941	Skewness	0.037862949
Range	799.41	Range	799.7
Minimum	0.22	Minimum	0.14
Maximum	799.63	Maximum	799.84
Sum	3703325.28	Sum	3703334.49
Count	9998	Count	9998

Table 2 Statistics of the Result

Average size of simulation is 100010(Cloudlet size), Considering the delay minimum time taken by a virtual machine in executing cloudlets is 0.14 ms and without any delay min-max will consume 0.11 ms and the maximum time taken by is almost similar to that of min-max whose difference is 0.21ms.

Cloudlet size	Min Max(ms)	Dynamic Min-Max (ms)
500	2	1.72
1000	4	3.45
1500	6	5.17
2000	8	7.41
2500	9.26	8.66
3000	11.11	11.11
3500	12.07	12.07
4000	13.79	16
4500	18	18
5000	20	18.52
5500	20.37	19
6000	22.22	22.22
6500	22.41	22.41
7000	24.14	28
7500	30	30
8000	32	29.63
8500	31.48	29.31
9000	33.33	33.33
9500	32.76	32.76

10000	34.48	40
10500	42	38.89
11000	44	37.93
11500	42.59	46
12000	44.44	41.38
12500	43.1	46.29
13000	44.83	52
13500	54	46.55
14000	56	51.85
14500	53.7	58
15000	55.56	51.72
15500	53.45	57.4
16000	55.17	55.17
16500	66	66
17000	68	62.96
17500	64.81	60.34
18000	66.66	72
18500	63.79	68.52
19000	65.52	65.52
19500	78	72.22
20000	80	80
20500	75.92	70.69
21000	77.77	84
21500	74.14	79.63
22000	75.86	75.86
22500	90	83.33
23000	92	79.31
23500	87.04	94
24000	88.89	96
24500	84.48	90.74
25000	86.21	86.21

Table 3 Comparison of Execution Time

The table gives the information about the cloudlet size, time taken to execute on both algorithms respectively.

- First row represents cloudlet size
- Second row represents time taken by min max algorithm
- Third row represents time taken by proposed algorithm

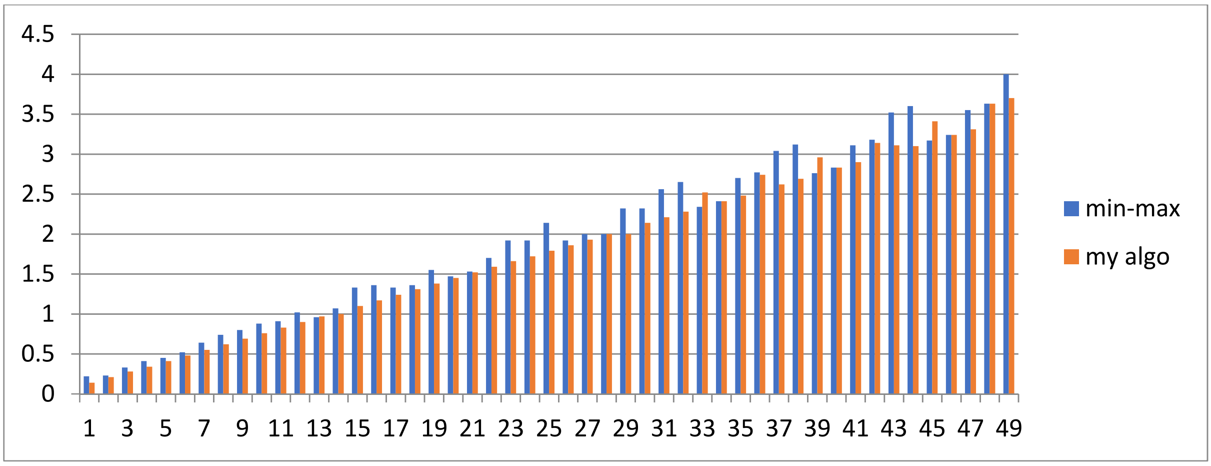

Figure 2: Execution time graph Min-Max vs Dynamic Min-Max

A sample of 50 tasks are taken from the simulation result, X-axis specifies the cloudlet Id and y-axis specifies time (ms) taken to execute that task

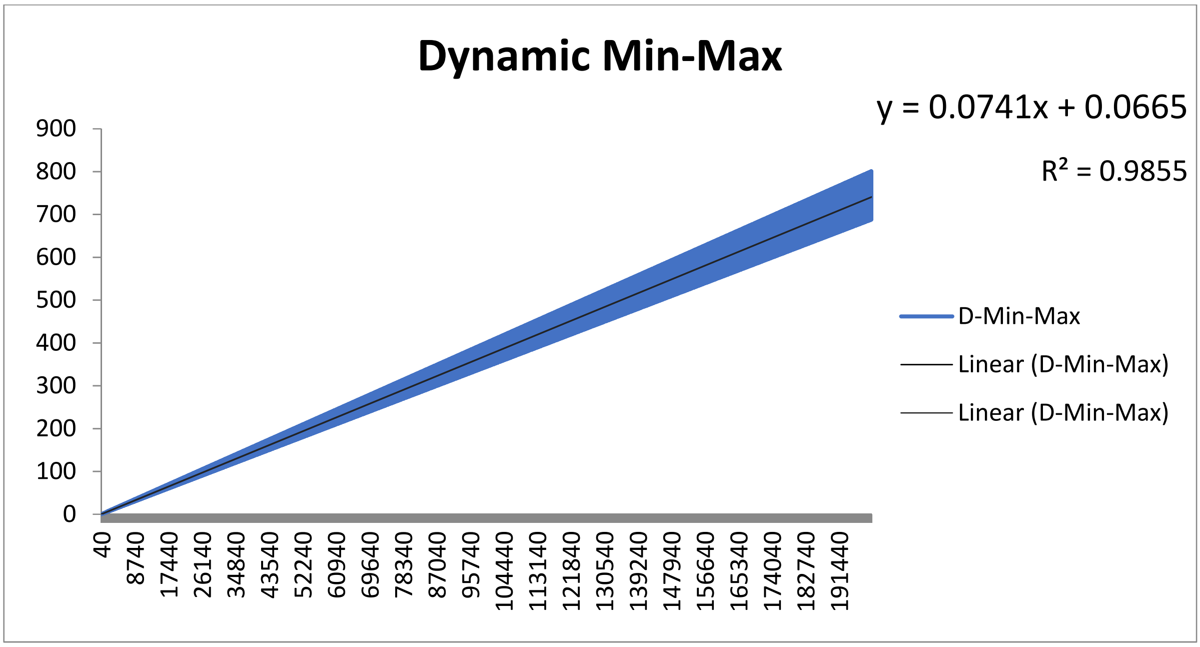

Figure 3: Regression trend line

Regression Analysis is a collection of statistical techniques or methods to estimate the relationship between a dependent and one or more independent variables.
In the proposed work, we have performed regression analysis considering the cloudlet size and the time taken to execute that and is given in “Eq. (1)”

$$Y=0.0741(x)-0.0665 \qquad (1)$$

The above “Eq. (1)” predicts the time taken to execute the task of size x
Y-axis represents time taken to execute cloudlet size of x instructions
X-axis represents instruction size.

Assuming Min-Max uses sorting algorithm to sort cloudlets based on their size before execution of task , We have to consider the time and space complexity of that algorithm assuming it uses best algorithms like merge sort and quicksort , both has the time complexity of O(nLogN)[average case] and space complexity of at least O(nlogn) . It is an additional burden to the processor to sort and then execute. When there is a large number of task executions requested at a time. The response time will affect the performance since the request are heterogeneous only time taken to compute the task is shown in the above graph

6.1 Inference:

Support vector machines (SVM) are supervised learning models used for examining the data for classification and regression analysis with related learning algorithms. Data scientists use them frequently to solve classification problems. SVM carries out classification by constructing hyper planes in a multidimensional space with discrete observations using distinct class labels. The ‘e1071’ package is used to implement SVM in R. The sVM() function in the e1071 package is used to implement the SVM supervised learning algorithm.
In this paper, SVM is used to devise an optimal hyperplane to correctly classify the response variable (virtual machine (VM_Class)) considering the predictor variable (Clouslet_Size) in the data set.
The model is devised as shown in Table 4.

```
install.packages("e1071")
library(e1071)
sVM1 <- sVM(VM_Class~., data=train_data, method="C-
classification", kernal="radial",
          gamma=0.1, cost=10, scale=F)
summary(sVM1)
```

Table 4 SVM Model Devised using R

```
 Call:
sVM(formula = VM_Class ~ ., data = train_data, method =
"C-classification", kernal = "radial",
   gamma = 0.1, cost = 10, scale = F)
Parameters:
 SVM-Type:  eps-regression
 SVM-Kernel:  radial
     cost:  10
     gamma:  0.1
     epsilon:  0.1
Number of Support Vectors:  3732
```

Table 5 The SVM Model summary

From Table 5 it is observed that kernel=”radial” (the default) is used for the multi-class classification problem. (VM_Classes are classified as 1,2 and 3). The values of gamma and cost are specified to find the best classification accuracy. To fetch useful information on how a model is trained, SVM’s summary() function can be used. There are 3732 different support vectors defined distributed across the classes 1, 2, and 3.

The predict() function is used as shown in Table 4, with the trained SVM model to perform predictions on the sample data set. The result is derived as a factor variable that holds the predicted classes for each observation in the sample data set. The “confusion matrix” is created for checking the accuracy of the model. The results are stated in Table 6.

```
prediction <- predict(sVM1, test_data)
cmat <- table(test_data$VM_Class, prediction)
prediction
 1          629
 2          1253
 3          618
```

Table 6 Predictions using Test Data Set

These results are helpful to decide how well the trained model will be useful to perform prediction using the sample data set. The prediction accuracy is calculated using "Eq. (1)" as follows
(629+1253+618)/ nrow (test_data) (2)
From "Eq. (2)" it is observed that the accuracy of the model is 99.3% which is very good.

7. Conclusion

Resource provisioning is a technique for allocating and scheduling resources based on the demand to ensure guaranteed performance for applications. The techniques used are selected in order to improve response time, performance. The Dynamic Min-Max algorithm provides a technique to handle requests dynamically as and when it comes as against the Min-Max algorithm which does batch processing. This algorithm can handle dynamic patterns in data arrivals and allocate resources accordingly. The support vector concept used here also provides a prediction model which can be used in future to make resource allocations. The Dynamic Min-Max also eliminates the overhead of sorting the tasks as against Min-Max algorithm. The prediction model used for future resource allocation has an accuracy of 99.3% which is ideal in any cloud environment.

References

[1] Sanjay Chakraborty, Nilotpal Choudhury,"A Study of a New Dynamic Load Balancing Approach in Cloud Environment",World Journal of Computer Application and Technology 4(3): 31-37, 2016.

[2] Priya Gupta, Makrand Samvatsar, Upendra Singh. "Cloud computing through dynamic resource allocation scheme", 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), 2017

[3] N. Susila, Dr. S.Chandramathi , "Energy Efficient Extended FCFS Load Balancing In Data Centers of Cloud" ,International Journal of Applied Engineering Research ISSN 0973-4562 Volume 11, Number 1 (2016) pp 599-605 © Research India Publications https://www.ripublication.com/ijaer16/ijaerv11n1_88.pdf.

[4] Abdulhussein Abdulmohson, Sudha Pelluri and Ramachandram Siranda ,"Energy Efficient Load Balancing of Virtual Machines in Cloud Environments", International Journal of Cloud-Computing and Super-Computing Vol.2, No.1 (2015), pp.21-34 http://dx.doi.org/10.21742/ijcs.2015.2.1.03

[5] Srinivas Sethi, Anupama Sahu ,Suvendu Kumar Jena , "Efficient load Balancing in Cloud Computing using Fuzzy Logic",IOSR Journal of Engineering (IOSRJEN) . ISSN: 2250-3021 Volume 2, Issue 7(July 2012), PP 65-71,www.iosrjen.org

[6] Zhang Qian, Ge Yufei, Liang Hong, Shi Jin, "Load Balancing Task Scheduling Algorithm based on Feedback Mechanism for Cloud Computing" ,International Journal of and Distributed Computing Vol.9, No.4(2016),pp.4152, a. http://dx.doi.org/10.14257/ijgdc.2016.9.4.04

[7] O. M. Elzeki,M. Z. Reshad,M. A. Elsoud , "Improved Max-Min Algorithm in Cloud Computing" ,International Journal of Computer Applications (0975 – 8887) Volume50 – No.12, July 2012 research.ijcaonline.org/volume50/number12/pxc3881009.pdf

[8] Sagar D. Girase, Mayank Sohani ,Suraj Patil,"Dynamic Resource Provisioning in Cloud Computing Environment using Priority based Virtual Machine's" 2014 IEEE International Conference on Advanced Communications (lCACCCT)

[9] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César A. F. De Rose, and Rajkumar Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms", Software: Practice and Experience, Volume 41, Issue 1, pages 23–50, January 2011.

[10] Raj Kumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility", Future Generation Computer Systems, Volume 25, Number 6, Pages: 599-616, ISSN: 0167-739X, Elsevier Science,Amsterdam.

[11] Huankai Chen,Professor Frank Wang,Dr Na Helian,Gbola Akanmu , Parallel Computing Technologies,"User-Priority Guided Min-Min Scheduling Algorithm For Load Balancing in Cloud Computing" , Parallel Computing Technologies (PARCOMPTECH), 2013 National Conference, http://ieeexplore.ieee.org/document/6621389/

[12] Sridharshini V, V.M.Sivagami ,"Energy-Aware Scheduling Using Workload Consolidation Techniques in Cloud Environment", International Journal of Computer Science and Engineering Communications Vol.3, Issue 3, 2015, Page.1141-1148

[13] Ilia Pietri, Maciej Malawski ,Gideon Juve , Ewa Deelman , Jarek Nabrzyski, Rizos Sakellariou , "Energy-Constrained Provisioning for Scientific Workflow Ensembles", 2013 IEEE Third International Conference on Cloud and Green Computing

[14] Dzmitry Kliazovich, Sisay T. Arzo, Fabrizio Granelli, Pascal Bouvry and Samee Ullah Khan, "e-STAB: Energy-Efficient Scheduling for Cloud Computing Applications with Traffic Load Balancing" 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE

[15] Kunduru Sravani , P.Rajendra Prasad, T.Madhu"A Unique Method for Energy Aware Load Balancing and Application Scaling in the Cloud Ecosystem", international journal of engineering technology and management research,Volume no 3 september(2016)

[16] Pooja Chauhan, Manjeet Gupta ," Energy Aware Cloud Computing Using Dynamic Voltage Frequency Scaling",International Journal of Computer Science And Technology Vol. 5, Issue 4, Oct - Dec 2014.

[17] R. Lee and B. Jeng "Load Balancing Tactics In Cloud" International Conference On Cyber Enabled Distributed Computing And Knowledge Discovery, (2011).

[18] N. J. Kansal, "Cloud Load Balancing Techniques: A Step Towards Green Computing", IJCSI International Journal Of Computer Science Issues, ISSN (Online): 1694-0814, vol. 9, Issue 1, no. 1, (2012) January, pp. 238-246.

[19] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt and A. Warfield,"Xen and the art of virtualization", in: Proceedings of the 19th ACM Symposium on Operating SystemsPrinciples, SOSP 2003, Bolton Landing, NY, USA, (2003), pp. 177.

[20] Mr. Jayant Adhikari, Prof. Sulabha Patil, "Load Balancing The Essential Factor In Cloud Computing", International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 1 Issue 10, December- 2012.

[21] Mr. Jayant Adhikari, Prof. Sulabha Patil, "Double Threshold Energy Aware Load Balancing In Cloud Computing", 4th IEEE International Conference on Computing, Communication and Networking Technology, 4-6 July 2013.

[22] oria Bidi, Zakaria Elberrichi. "Using Penguins Search Optimization Algorithm for Best Features Selection for Biomedical Data Classification", International Journal of Organizational and Collective Intelligence, 2017