Scheduling of Heterogeneous tasks in cloud computing using Multi Queue (MQ) Approach

Pooja[1], Dr Sanjay Tyagi[2]

[1]*M. Tech. Scholar, Department of Computer Science & Applications, Kurukshetra University, Haryana, India*
[2]*Assistant Professor, Department of Computer Science & Application, Kurukshetra University, Haryana, India*

Abstract- *Cloud computing is increasing very fast & with its rapid increment, the requirement of computations is also increasing in cloud environment. There are multiple issues that exist in cloud environment like quality of services (QoS) requirement, minimum energy consumption and scheduling of tasks. Number of task scheduling algorithms exist in cloud computing, which schedule the tasks to available resources in easy way. In this paper, Multi Queue (MQ) task scheduling algorithm has been purposed to improve the performance of system. Multi Queue (MQ) scheduling algorithm overcomes the drawbacks of existing Round Robin and Weighted Round Robin algorithms. CloudSim toolkit has been used to simulate the proposed work. Experiment results show that the proposed Multi Queue (MQ) scheduling algorithm performs better as compared to exiting Round Robin (RR) and Weighted Round Robin (WRR) algorithms.*

Keywords: Cloud Computing, Cloud Service provider, Round Robin (RR) Algorithm, Task Scheduling, Virtual Machine, Weighted Round Robin (WRR) Algorithm

1. Introduction

Cloud computing is internet based technology, which provides on demand resources, software and other information to computer. Big advantage of cloud computing is that it provides total low cost, greater flexibility and fast service [1]. Cloud service provider maintains the cloud computing infrastructure and services. One major drawback of cloud computing is that it works in dynamic environment. Cloud computing offers three types of services viz. Infrastructure as a service (IaaS), Platform as a service (PaaS) and Software as a service (SaaS) [2]. Deployment of task scheduling plays an important role in cloud computing.

Scheduling of task is a big challenge in cloud computing and it is NP-hard optimization problem. Tasks are scheduled according to user requirements. Over utilization and underutilization states in task scheduling should be minimized because they affect the overall response time and throughput. The main aim of scheduling is to map the incoming tasks to the VMs according to scheduling policy. For successful execution of these tasks number of virtual machines is deployed, since there are number of virtual machines that can be created and destroyed inside physical machine [3]. Task scheduling not only maintains the throughput and response time, but it is also helpful for providing good QoS (Quality of Service) by maintaining the conditions of SLA's (Service Level Agreement).

Three main different phases of scheduling in cloud computing are:

First Phase is resource discovery, a list of resources is generated by this phase. In second phase, information about all resources are gathered and the best resource according to application requirement is chosen. Finally in third phase, execution of tasks takes place, which includes cleanup and file staging [4]. Basic steps used in these three phases are shown in figure 1.

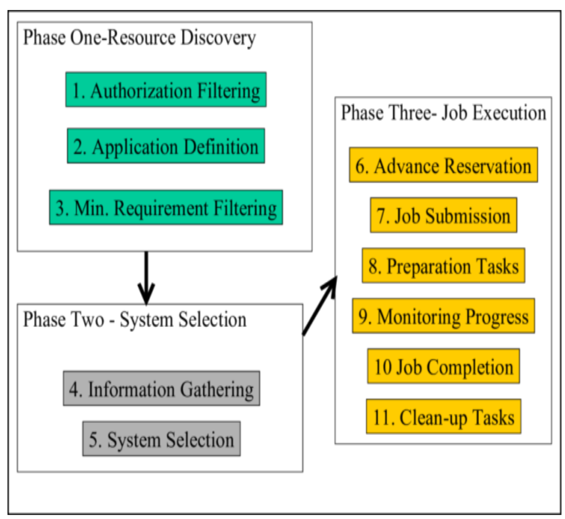

Figure - 1: Three-Phase Architecture for cloud Scheduling

In task scheduling, makespan and resource utilization are two most important parameters used. In order to obtain better results, minimization of makespan and maximization of resources utilization should take place. Both parameters are equally important in task scheduling to make the balance [5].

Round Robin (RR) is one of the popular task scheduling algorithms in cloud environment. It is basically depending on the principle of time slice, also known as quantum time (QT). Here time is divided into multiple segments or time slice and a particular time slice is given to each resource. Sum of all time slices is known as RR turn [6]. Each resource completes the execution of incoming task in a particular time slice and all the incoming tasks are sequentially executed on available VM according to RR fashion. If task is not completely executed in a given time slice, then it has to wait for the next RR turn and goes in waiting state. In next RR turn, task is completely executed and due to this waiting time, number of context switching increases and too much overhead occurs in the system [7]. This process in repeated until there are no more tasks in ready queue.

Weighted Round Robin (WRR) is the advance version of Round Robin (RR) scheduling algorithm. It overcomes the disadvantages of basic Round Robin algorithm and also improves the performance of system in cloud environment. In Weighted Round Robin algorithm, firstly weight is assigned to each resource which shows capability of resources. After that, these resources are sorted in descending order. All incoming tasks are executed on available resources in RR order according to the capability of VM. Resources having highest weight receive more request than the resource having small weight [8] [9]. Weighted Round Robin consume less time as compared to Round Robin scheduling algorithm [10].

In this paper, Multi Queue task scheduling algorithm has been proposed. As its name suggests, it is a scheduling algorithm that overcomes disadvantages of basic round robin (RR) and weighted round robin (WRR) algorithms. It gives better Makespan, Load Balancing level and Resource Utilization in most of the cases when it is compared with Round Robin (RR) and Weighted Round Robin (WRR).

The remaining paper has been organized as follows: Section 1 gives the brief introduction of cloud computing and two basic task scheduling algorithms, Section 2 shows the related work, Section 3 presents the proposed Multi Queue (MQ) task scheduling algorithm. Next section i.e. section 4 discusses the experiment results & analysis of proposed algorithm and its comparison with the existing RR and WRR scheduling algorithms. This section is followed by section 5, which concludes the paper.

2. Related Work

To overcome the power consumption problem that occurs in basic round robin algorithm, an Enhanced Weighted Round Robin (EWRR) was developed by Abdulaziz Alnowiser, et al. [11]. It was based on modified round robin algorithm with VM reuse and VM migration. In order to maximize the energy saving in data center, EWRR algorithm used Dynamic Voltage and Frequency Scaling technology. DVFS technology was used to adjust the VM frequency which was dependent on workload and it also minimized the energy consumption. Different priorities were assigned to different weighting coefficient queue using EWRR algorithm and on the basis of these priorities, tasks were assigned to VM.

For scheduling the task, round robin algorithm was mainly used and its performance is highly dependent on Quantum size. There is a problem of performance degradation with respect to average waiting time (AWT), average turnaround time (ATT) and number of context switches (NCS) occurring in round robin scheduling algorithm. To overcome this problem, an improved Dynamic Round Robin Scheduling Algorithm Based on a Variant Quantum Time was developed by Ahmed Alsheikhy, et al. [7]. The main aim of this algorithm was to improve the overall system performance and maximize the throughput of system with minimization of average waiting time, average turnaround time and number of context switching. By choosing large quantum size, maximum number of task completed their execution with minimization of overhead occurring during context switching.

Alternating Median Based Round Robin scheduling algorithm was purposed by Salman Arif, et al. [12]. It overcomes the problem of reducing either of response time or number of context switches and waiting time during scheduling of tasks. Using AMBRR algorithm, first priority is given to the reduction of the number of context switches & waiting time and second priority to the reduction of response time. On the basis of two time quanta which are used in AMBRR algorithm, scheduling of task takes place and these two time quanta remain fixed throughout the process. The value of first quanta is equal to the median of burst time of all processes and value of second quanta is equal to the difference between highest values of burst time and median of burst time of processes. In cloud computing, QoS's-aware task scheduling is NP-hard optimization problem. To overcome this NP-hard problem, Template-based Genetic Algorithm (TBGA) with QoS's constraints was developed by Xiaodong Sheng, et al. [13]. In this algorithm, firstly template of task is calculated; after that using this template task are combined into multiple subset and finally the subset of tasks are allocated to resources using genetic algorithm. Size of tasks is made equal to the processor's template using TBGA because value of gene is related to the size of processors' template. TBGA minimizes the makespan and gives better result as compared to other scheduling algorithms.

3. Proposed algorithm

In cloud computing environment, number of task scheduling algorithms exists. RR and WRR are two of them having some advantages and disadvantages. In round robin algorithm, sequence of heterogeneous task is executed on available resources. Time is already fixed for all the tasks that are ready to execute. When task begin its execution, Quantum Time (QT) is assigned to the task. If the task can't complete its execution in a given QT, these tasks are blocked and placed at the end of the list. This process is repeated till no task is remaining in the list. In this algorithm, fewer QT increases overhead and high Quantum time increases average waiting time and turn-around time [14]. In weighted round robin algorithm, firstly weight is assigned to the resources, after that resource are sorted in descending order according to their capability with the help of collection sorting. It allocates incoming tasks to the available resources in Round Robin order [15]. WRR consumes less time as compared to RR, but there exists some problems like maximum makespan, low resources utilization and improper load balancing.

To overcome these problems, Multi Queue (MQ) task scheduling has been proposed. In proposed algorithm, tasks and resources are split into two queues: small & large and slow & fast. MQ algorithm provides better results in most of the cases. It minimizes the makespan of tasks, maximizes resources utilization ratio and balances the load equally between all the available resources.

Multi Queue scheduling algorithm uses the advantages of Round Robin and Weighted Round scheduling algorithms and overcomes their drawbacks. First of all, it calculates the Completion Time (CT_i) of each task in metaSet buffer of size n, and then it calculates Average Task length (AvgTL). After that, list of tasks is splitted into two parts: small and large. Then AvgTL is compared with each task-len (i), if task-len (i) is shorter than AvgTL, then that task i is added to small queue otherwise it is added to large queue. In next step, calculate AvgMIPS. After that, list of resources of size m is splitted into two parts: slow and fast. Then, AvgMIPS is compared with each resource MIPS, if MIPS is shorter than AvgMIPS, that resource j will be added to slow resource otherwise added to fast one. This process repeats until all the tasks in metaSet buffer and resources have been compared and it gets empty. After completing this process, all small tasks are scheduled on slow virtual machine and large tasks are scheduled on fast virtual machine.

Flow chart of this complete process has been shown in figure 2:

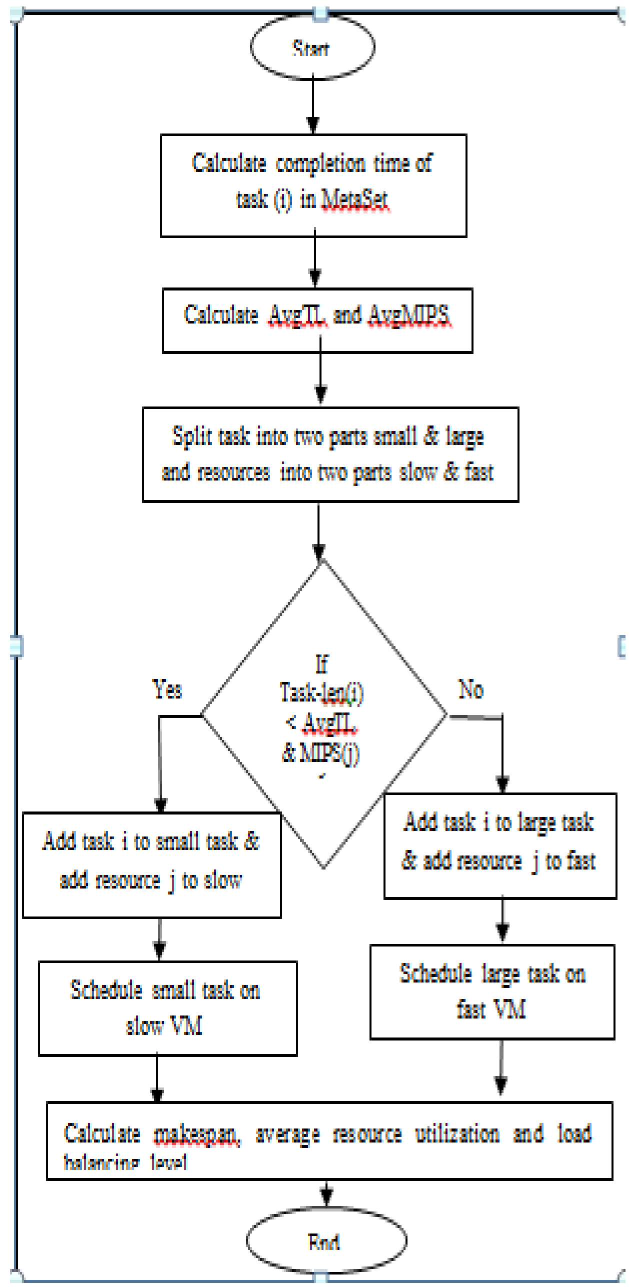

Figure - 2: Flow Chart of Proposed Multi Queue (MQ) scheduling algorithm

Pseudo Code for Multi Queue (MQ) task scheduling Algorithm:

Store all tasks & resources in metaSet buffer of size 'n' and resource buffer of size 'm'.

for all tasks t_i & resource R_j
$CT_{ij}=ET_{ij} + r_j$
end
Calculate the AvgTL=
AvgTL = $\sum_{i=1}^{n} task - len\,(i)$/n
Split list of task into two parts- small and large
While MetaSet is not empty
Start
if (task-len (i) > AvgTL)
Add Task i to large
else
Add Task i to small
end while
Calculate the AvgMIPS=
AvgMIPS = $\sum_{j=1}^{m} MIPS/m$
Split the resources into two parts- slow and fast
While Resource buffer is not empty
Start
If (MIPS < AvgMIPS)
Add resource j to slow
Else
Add resource j to fast
End While
Schedule the small task on slow VM
Schedule the large task on fast VM
End
End

Here, i represents the tasks present in the MetaSet buffer, j represents the number of resources and r defines a particular resource. CT_{ij} has been used for completion time of task i on resource j, and ET_{ij} defines the execution time of task i on resource j.

4. EXPERIMENTAL RESULTS & ANALYSIS

For simulating the proposed algorithm, CloudSim toolkit has been used. Some important features of CloudSim are:

- It is helpful in modeling virtualized resource configuration, used in simulation, e.g. their RAM size, bandwidth, MIPS rate etc.
- It supports large-scale simulation experiment.
- No upper limit is provided by CloudSim tool for number of resources and tasks used in simulation process [16].

The experiments have been carried out to compare the performance of the proposed algorithm with existing algorithms on the basis of three performance matrix: Makespan, Average Resource Utilization Ratio and Load balancing level.

Makespan: It is the total execution time in which task get scheduled or completely executed. For better performance of cloud system, makespan always should be low [17].

Average Resource Utilization: It can be defined as the complete utilization of each resource present in cloud environment. For better performance of cloud system, average resource utilization ratio should be high.

Load balancing level: Load balancing is a major issue in cloud computing. Due to improper resource utilization, it is very difficult to balance the load equally on all resources [18]. For better performance of cloud system, Load balancing level should be high.
Performance results have been shown below for these parameters. Here, the proposed algorithm has been compared with Round Robin and Weighted Round Robin scheduling algorithm. In this scenario, number of resources remain constant (R=10) and number of tasks have been changed for performance evaluation.
The results for makespan performance metric have been represented in a tabular form as well as in graphical form, where four different numbers of task sizes have been taken and the simulation has been performed on each task size to obtain the result.

Table - 1: Makespan Analysis (in ms)

Number of tasks	Round Robin	Weighted Round Robin	Multi Queue
50	50.10	50.21	33.10
100	100.10	100.21	67.21
200	200.10	200.21	137.21
300	300.10	300.42	200.42

Table-1 shows the makespan values of Round Robin, Weighted Round Robin and proposed algorithm with four different scenarios. First, 50 tasks have been taken and after that performance is shown on high load for which 100, 200 and 300 tasks have been taken.

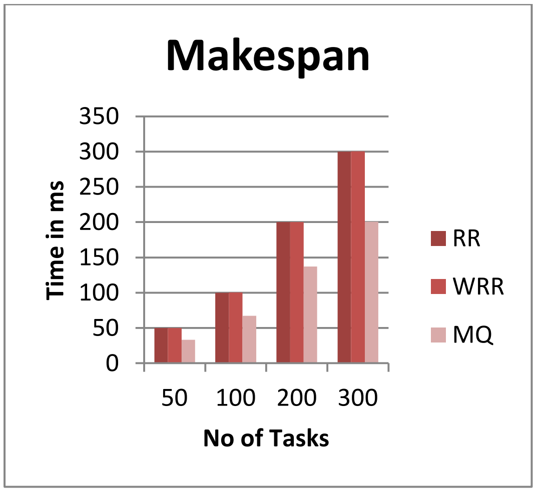

Figure - 3: Comparison of makespan (50, 100, 200, 300 tasks)

Figure 3 represents the comparison of makespan between three different tasks scheduling algorithms. Multi Queue (MQ) algorithm gives better result as compared to existing Round Robin and Weighted Round Robin both algorithms in all the cases.

Table 2: Average Resource Utilization Analysis

Number of tasks	Round Robin	Weighted Round Robin	Multi Queue
50	35.43	39.49	52.16
100	35.46	39.56	51.13
200	35.48	39.61	51.07
300	35.49	39.59	52.57

Table-2 represents the average resource utilization rate of Round Robin, Weighted Round Robin and the proposed algorithm in four different scenarios, first with 50 tasks after that performance have been shown on high load using 100, 200 and 300 cloudlets.

Figure 4 represents the comparison of average resource utilization rate between three different tasks scheduling algorithms. Multi Queue (MQ) algorithm gives better result as compared to existing Round Robin and Weighted Round Robin both algorithms in all the scenarios.

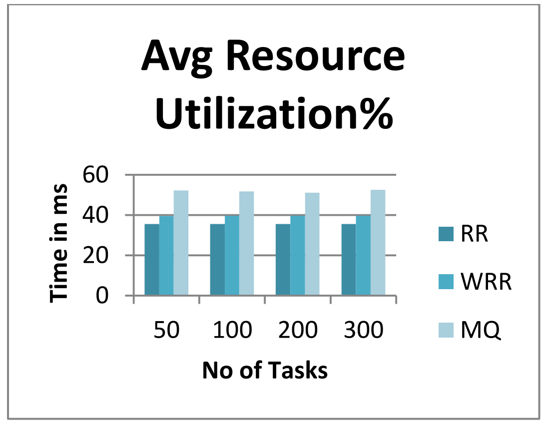

Figure - 4: Comparison of Avg Resource Utilization (50, 100, 200, 300 tasks)

Table 3: Load Balancing Level Analysis

Number of tasks	Round Robin	Weighted Round Robin	Multi Queue
50	21.72	19.91	53.11
100	21.72	19.94	52.13
200	21.72	20.08	50.66
300	21.72	20.08	49.90

Table-3 represents the load balancing level values of round robin, weighted round robin and proposed algorithm within four different scenarios, first with 50 tasks and after that performance is shown on high load for which 100, 200 and 300 tasks have been taken.

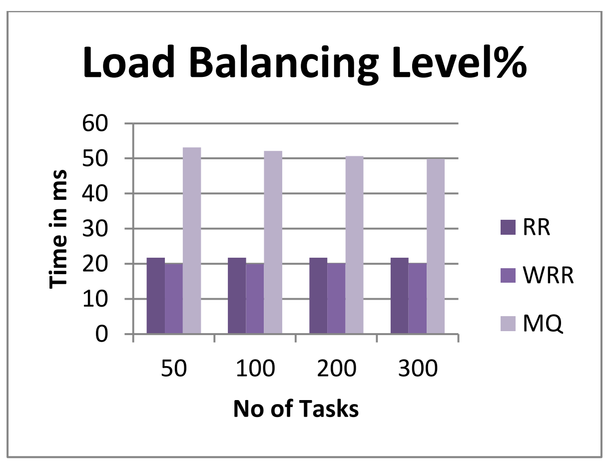

Figure - 5: Comparison of load balancing level (50,100, 200, 300 tasks)

Figure 5 represents the comparison of load balancing level between three different task scheduling algorithms. Multi Queue (MQ) algorithm gives better results as compared to existing Round Robin and Weighted Round Robin both algorithms in all the cases.

Hence all the above results have shown that Multi Queue task scheduling algorithm performs better for makespan, average resource utilization and load balancing level performance metrics and gives better results as compared to Round Robin and Weighted Round Robin scheduling algorithms.

5. CONCLUSION AND FUTURE WORK

Scheduling of tasks in cloud environment is one of the big issues. There are numbers of task scheduling algorithm. In this paper, a new effective and efficient task scheduling algorithm has been proposed. Multi Queue (MQ) scheduling algorithm overcomes the drawbacks of Round Robin and Weighted Round Robin scheduling algorithms and gives better makespan, average resource utilization rate and load balancing level as compared to existing algorithms. CloudSim toolkit has been used for simulating the results. In future work using the concept of min-min algorithm, performance can be increased.

REFERENCES

[1] M. B. M. Varghese and D. R. J. S. Raj, "A Survey on Variants of Genetic Algorithm for Scheduling Workflow of Task," in Second International Conference on Science Technology Engineering and Management (ICONSTEM), IEEE, pp. 489-492, 2016.

[2] O. Kaneria and R. K. Banyal, "Analysis ans Improvement of Load Balancing in Cloud Computing," International Conference on ICT in Business Industry & Government (ICTBIG), IEEE, pp. 1-5, 18-19 Nov 2016.

[3] M. Agarwal and D. G. M. S. Srivastava, "A Genetic Algorithm inspired task scheduling in Cloud Computing," in International Conference on Computing and Automation (ICCCC2016), IEEE, pp. 364-367, 2016.

[4] S.Devipriya and C.Ramesh, "Improved Max-Min Heuristic Model For Task Scheduling In Cloud," in International Conference on Green Computing, Communication and Conservation of Energy (ICGCE),IEEE, pp. 883-888, 2013.

[5] B. Taneja, "An Empirical Study of Most Fit, Max-Min and Priority Task Scheduling Algorithm in Cloud Computing," in International Conference on Computing, Communication and Automation (ICCCA2015), IEEE, pp. 664-667, 2015.

[6] R. Racu, L. Li, R. Henia, A. Hamann and R. Ernst, "Improved Response Time Analysis of Task Scheduling under Preemptive Round Robin," in (ESWEEK) Embedded Systems Week, pp. 79-84 2007.

[7] A. Alsheikhy, R. Ammar and R. Elfouly, "An Improved Dynamic Round Robin Scheduling Algorithm Based on a Variant Quantum Time," in IEEE, pp. 98-104, 2015.

[8] P. Samal and P. Mishra, "Analysis of Variants in Round Robin Algorithm for load balancing in Cloud Computing," International Journal of Computer Science and Information Technology (IJCSIT), vol. 4, no. 3, pp. 416-419, 2013.

[9] X. Zongyu and W. Xingxuan, "A Predictive Modified Round Robin Scheduling algorithm for web server clusters," in National Nature Science Foundation of China (NSFC), pp. 5804-5808, 2015.

[10] S. Supreeth and S. Biradar, "Scheduling Virtual Machine for Load balancing in Cloud Computing Platform," International Journal of Science and Research (IJSR), vol. 2, no. 6, pp. 437-441, June2013.

[11] A. Alnowiser, E. Aldhahri and A. Alahmadi, "Enhanced Weighted Round Robin (EWRR) Scheduling with DVFS Technology in Cloud," in International Conference on Computational Science and Computational Intelligence,IEEE, pp. 320-326, 2014.

[12] S. Arif, N. Ghaffar and A. Javed, "Implementation of Alternating Median Based Round Robin Scheduling Algorithm," in International Conference on Computer and Information Technology, IEEE, pp. 154-160, 2016.

[13] X. Sheng and Q. Li, "Template-based Genetic Algorithm for QOS-aware Task Scheduling in Cloud Computing," in International Conference on Advanced Cloud and Big Data,IEEE, pp. 25-29, 2016.

[14] Pooja and Dr. S. Tyagi, "Task Scheduling in Public Cloud: A Review," International Journal of Engineering Science and Technology (IJEST), vol. 9, no. 5, pp. 403 - 407, May 2017

[15] S. Dave and P. Meheta, "Utilizing Round Robin Concept for Load Balancing Algorithm at Virtual Machine Level in Cloud Environment, "International Journal of Computer Application", vol. 94, no. 4, pp. 23-29, May2014.

[16] N. Sharma, Dr. S. Tyagi and S. Atri, "HYMM: A New Heuristic in Cloud Computing," International

Research Journal of Engineering and Technology (IRJET), vol. 4, no. 5, pp. 3520-3526, May 2017.

[17] G. Pate and R. Mehta, "A Survey on Various Task Scheduling Algorithm in Cloud Computing," International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 3, no. 3, pp. 715-717, March2017.

[18] R. Kaur and P. Luthra, "Load Balancing in Cloud System using Max Min and Min Min Algorithm," International Journal of Computer Applications (0975 – 8887) National Conference on Emerging Trends in Computer Technology (NCETCT-2014), pp. 31-34, 2014.

.