International Journal of Engineering Research
Volume No.6, Issue No.4, pp :228-232
ISSN:2319-6890 (online),2347-5013(print)
1 April. 2017

# A Survey on Modern Load Balancing Algorithms in Cloud Computing

**Shashi Mehar, Hansa Acharya**
Department of CSE, RITS, Bhopal M.P. India
shashimeher2010@gmail.com, hansaacharya@gmail.com

***Abstract— Cloud computing is a new and innovative perspective for large scale parallel and distributed computing. The dependence of user or load on the cloud is growing enormously with the enlargement of new applications. Load balancing is a significant area of cloud computing environment which ensures that all connected devices or processors carry out same amount of work in equal time. With an aim to make cloud resources and services accessible to the cloud user easily and conveniently, different algorithms and models for load balancing in cloud computing is being developed. This paper aims to deliver an ordered and comprehensive summary of the research work carried out in the field of load balancing algorithms in cloud computing. This paper surveys the state of the art load balancing tools and techniques. Existing approaches are discussed and analyzed with an aim to provide load balancing in a fair way this paper also provides the comparative analysis of performance of different existing load balancing algorithms in cloud environment.***

*Keywords*— **Cloud Computing, Load Balancing Algorithms, Load Balancers, Task Scheduling, Resource Allocation, Minimum Execution Time.**

## I. INTRODUCTION

Cloud computing delivers elastic or flexible means to retain data and files that requires virtualization, distributed computing, and internet services. It additionally has many components like client and distributed servers (see Figure 1). The main motive of cloud computing is to deliver maximum services at minimum cost anywhere at any time. Currently, there are more than hundred millions of computing devices are connected to the Internet. Such devices put forward their request and receive the response devoid of any delay. Different devices such as tablet, PCs or laptops are connected and may access the data/information from a cloud at any specified time [1]. The most important objectives of cloud is to minimize cost, increase response time, deliver high performance, hence Cloud is also termed as a pool of resources or services. Load may be described in different types like, CPU load, network load, memory capacity problems etc. In the terms of cloud computing, load balancing is concept to share load of virtual machines across all nodes (end user devices) to improve resources, service utilization and provides high satisfaction to users. Due to load distribution or sharing, each and every node can work efficiently; data files can be acknowledged and sent without any delay. The dynamic load balancing technology uses system information while sharing the load. A dynamic scheme is quite flexible and fault tolerant [2]. Load balancing technique enables advance network services and assets or resources for improved response and performance. A number of techniques are used to stabilize cloud data among distributed nodes. All users' load is handled by cloud service provider for smooth provisioning of services.

Load balancing is typically functional on huge traffic of data to distribute work to virtualized servers. Highly developed architectures of cloud are adopted to acquire speed and efficiency. There is a number of uniqueness of load balancing such as: even division of load among the entire nodes, to achieve user satisfaction, enhance overall performance of system, reduction in response time, and services to acquire complete utilization of resources. As an example, if any one application is developed on cloud and hundreds of users are expected to access it at same time. Consequently, response time to hundred users will be very slow and servers will become very busy that results in poor response and hence not satisfactory to users. If load balancing algorithm is applied on cloud application, then load or submitted jobs will be distributed at all nodes and hence high performance and better response will be achieved[3,4].

The course of action in which the load is distributed among different nodes of distributed cloud architecture is called load balancing in cloud computing. Plenty of efforts have been done to handle load in order to enhance performance and avoid over utilization of resources. Various load balancing algorithms have been discussed including round robin (RR), Min-Min, Max-Min etc. Load balancing algorithm are divided in two main categories, namely static and dynamic [4].

In static algorithms decision about load balancing is made at compile time. These are limited to the environment where load variations are few. These algorithms are not dependent upon the present condition of system. A static load balancer algorithm divides the traffic equally among the servers. It does not use the system information while distributing the load and is less complex. Dynamic algorithm is based on the different properties of the nodes such as capabilities and network bandwidth. This need constant check of the node and are usually difficult to implement [4].

If we apply load balancing on our application, then work will be distributed at other nodes and we can get high performance and better response [5]. The existing survey does not critically discuss the available tools and techniques that are used in cloud computing. In this paper, we provide a comprehensive overview of interactive load balancing algorithms in cloud computing. Each algorithm addresses different problems from different

ISSN:2319-6890 (online),2347-5013(print)
1 April. 2017

aspects and provides different solutions. Some limitations of existing algorithms are performance issue, larger processing time, starvation and limited to the environment where load variations are few etc. A good load balancing algorithm should avoid the over loading of one node.


**Figure 1. Cloud Computing Architecture [1]**

The aim is to evaluate the performance of the cloud computing load balancing algorithms. The rest of the paper is organized as follows. In section II, we compare review different load balancing algorithms. In Section III, the performance evaluation of different cloud computing algorithms have been discussed and evaluated. Our discussion and findings are summarized and the paper is concluded in section IV.

## II. RELATED WORK

This section describes various existing work to be done in the field of load balancing or task scheduling in cloud computing environment. The load balancing in cloud computing systems is really a major challenge. Load balancing is one of the major issues of cloud computing which involves the dividing the total load equally [6]. So, the throughput of the system is to be increased and decrease the response time.

In [7] author proposed a modified Min-Min load balancing algorithm for static Meta-Task scheduling. This algorithm choose the tasks with maximum completion time and assign it to applicable resources to provide better makespan and utilization of resources with efficiency. This work execute in 2 stages using 2 existing algorithms i.e. Min-Min algorithm and Max-Min algorithm. 1st Min-Min strategy is applied and tasks are rescheduled to use unutilized resources effectively.

In [8], author proposed load balancing algorithm based on soft computing. For allocation of incoming tasks to the virtual machines(VMs) author proposed a local optimization approach based on Stochastic Hill climbing.

In [9] author proposed an algorithm for cloud environment which is based on energy optimization methods that could automatically assign resources. The proposed work presented an automatic procedure to find the appropriate CPU frequency, main memory's mode or speed. Algorithm had also presented elastic distributed monitoring software.

In [10] author presented a work that presented that scheduling of tasks are done by combining network awareness and energy efficiency. This work satisfies QoS requirements and improves job performance. Algorithm reduces the number of computing servers and avoids hotspots. Network awareness is obtained by using feedback channels from the main network switches. This method has less computational and memory overhead.

In [11] an optimized scheduling strategy is implemented to reduce power consumption along with satisfying task response time constraints during scheduling. It is a greedy approach which selects minimum number of most efficient servers for scheduling. The tasks are heterogeneous in nature so that they constitute different energy consumption levels and have various task response times. Optimal assignment is based on minimum energy consumption and minimum completion time of a task on a particular machine.

In [12] author had proposed an enhanced scheduling algorithm based on cost metric to schedule tasks. The cost of assignment varies for task to task dependent on their complexity. The algorithm takes into consideration the cost of resource and processing capability. Algorithm group tasks based on the processing capacity and select best resource to schedule for reducing cost. As compared to other scheduling algorithms, this algorithm reduces the makespan and the processing cost.

In [13] author introduces an algorithm that considers task's priority for scheduling. Task are sorted based on priorities which is assigned on the basis of different attributes of the tasks. They are assigned on VM that results in the enhanced completion time. Hence, performance of this algorithm shows better result by having better completion time.

In [14] author proposed based on the divisible load theory which aims to reduce the overall processing time of the tasks. Homogeneous processors are used here for that the load fractions and processing time for every task are calculated. The divisible load is partitioned off among completely different servers and therefore it permits the quickest completion of the tasks at intervals of a short period of time. This technique improves the cloud providers benefit additionally as quality of service. This technique results good in terms of performance, total cost, delay cost, efficiency in comparison to different random strategies.

In [15] author projected an algorithm that may be a modification done on the improved max-min algorithm. It's based on the expected execution time during which it assigns a task with average execution time on the machine which supplies minimum completion time. The largest task determines the makespan of the system and generally it's going to be large then it'll will increase the makespan of the system and make load imbalance. Therefore rather than selecting largest task they select average largest task or nearest average largest task. This technique produces additional truthful load balancing and makespan than improved max-min method.

ISSN:2319-6890 (online),2347-5013(print)
1 April. 2017

In [16] author had given an enhanced scheduling algorithm after analyzing the conventional algorithms that are based on user's priority and task length. High prioritized tasks are not specified any special importance when they arrive. The proposed technique considers all of these factors and designed an algorithm that assigns a credit value to all submitted tasks and further schedule tasks with respect to assigned credit.

## III. PERFORMANCE EVALUATION OF EXISTING TECHNOLOGIES

Table I compares different type of load balancing scenarios in cloud computing environment. It specifies the knowledge base, usage and drawbacks of each type of algorithm and issues addressed by these algorithms.

**Table I Different Cloud Computing Load Balancing Scenario**

| Algorithm | Description | Challenges Faced |
|---|---|---|
| **Static** | Former information is required about every node and user requirements. | Response time<br>Resource utilization<br>Scalability<br>Energy Utilization<br>Makespan<br>Throughput or Performance |
| **Dynamic** | According to regular changing user requirement it is necessary to monitor each and every node to analyze VM's at run time. | Location of VM to which load is transferred from overloaded VM.<br>Information Retrieval and processing.<br>Load calculation.<br>Throughput |
| **Centralized** | Single node or VM is responsible for maintenance of entire network status and updation of status time to time. | Threshold<br>Throughput<br>Failure<br>Communication overhead among central server and processors. |
| **Distributed** | Network is divided into several processors and each processor is responsible for balancing load and maintain their own local database for making effective load balancing decisions. | Assigning of processor to balance load in the network.<br>Migration time.<br>Interprocessor communication.<br>Throughput<br>Fault tolerance |
| **Hierarchical** | A level of hierarchy is maintained and all information are communicated from lower level nodes to higher level nodes to enhance network performance. | Threshold policies<br>Information exchange criteria<br>Selection of nodes at different levels of network<br>Failure intensity<br>Performance<br>Migration time |

**Table II. Comparative Study of Existing Load Balancing Algorithms**

| Algorithm | RT | THR | OH | SP | FT |
|---|---|---|---|---|---|
| Min-Min Algorithm | Fast | High | High | Fast | No |
| Max-Min Algorithm | Fast | High | High | Slow | No |
| Priority Based Algorithm | Low | Avg | High | Avg | No |
| Round Robin Based | Fast | High | High | Avg | No |
| Dynamic load Balancing | Slow | High | High | Fast | Yes |
| Ant Colony | Slow | High | High | Fast | N/A |
| Throttled load Balancing | Fast | Low | Low | Fast | Yes |

Every research has given contribution in enhancement of performance of the cloud services but also have a number of limitations, a few of them are listed below. Some of existing algorithms are discussed in Table II.

Below are the some abbreviations used in Table II as:

RT = Response Time, THR= Throughput, OH= Overhead, SP= speed, FT= Fault Tolerance. Some of the existing techniques proposed in cloud computing are discussed below:

### *A. Priority Based Modified Throttled Algorithm in Cloud Computing (PMTA)*

Proposed an algorithm, which adds a new feature like priority basis service allocation of each user request. Determining the priority of a request, the task allocated to a Virtual Machines. A Switching queue has proposed to hold the request which has been removed temporarily from the V.M. due to the arrival of higher priority request. The waiting request resumed the execution after

ISSN:2319-6890 (online),2347-5013(print)
1 April. 2017

completion of the of higher priority task[17]. The results of PMTA is given below as:

| Algorithm | Avg(ms) | Min(ms) | Max(ms) |
|---|---|---|---|
| Round Robin | 0.35 | 0.02 | 0.61 |
| Throttled Load Balancing | 0.1.1 | 0.019 | 0.114 |
| PMTA | 0.065 | 0.02 | 0.105 |

*B. NBST Algorithm: A load balancing algorithm in cloud computing*

An algorithm is proposed to balance load on cloud based on arrangement of resources, according to processing speed for virtual machines and then allocating cloudlets to the resources according to their processing requirement. This algorithm allocates the resources in such a manner that job requiring less processing are not allocated to the machines with high processing power and vice versa[18]. The results of NBST is given below as:

| Algorithm | Total Execution Speed (MIPS) |
|---|---|
| NBST | 35 |
| FCFS | 60 |

*C. Modified Round Robin Algorithm for Resource Allocation in Cloud Computing*

Modified round robin resource allocation algorithm is proposed using dynamic time quantum instead of fixed time quantum, which satisfies customer demands by reducing the waiting time [19]. The results of modified round robin is given below as:

| Algorithm | Average Waiting Time | Average Turn Around Time |
|---|---|---|
| Round Robin | 4.5 ms | 18 ms |
| Modified Round Robin | 2.1 ms | 17.5ms |

*D. Grouped tasks scheduling algorithm based on QoS in cloud computing network*

This paper proposes the grouped tasks scheduling (GTS) algorithm that is used to schedule tasks in cloud computing environment to satisfy user's needs. The GTS algorithm divide tasks into five groups on the basis of same attributes such as type of user, type of task, task size, and latency of task. After assigning each tasks right group, the algorithm starts scheduling these tasks into available VMs. Scheduling is done in 2 steps: 1st step decides which group will be scheduled first. 2nd step decides which task inside the chosen group will be scheduled first. GTS is dependent on the execution time of task so the task having minimum execution time will be scheduled first [20]. The results of GTS are given below as:

| Algorithm | Average Execution Time at 100 services |
|---|---|
| Min-Min | 51.56 ms |
| GTS | 51.24 ms |

## IV. PROPOSED METHODOLOGY

The proposed approach is based on two attributes of tasks submitted for scheduling purpose: Task Length and User Priority.

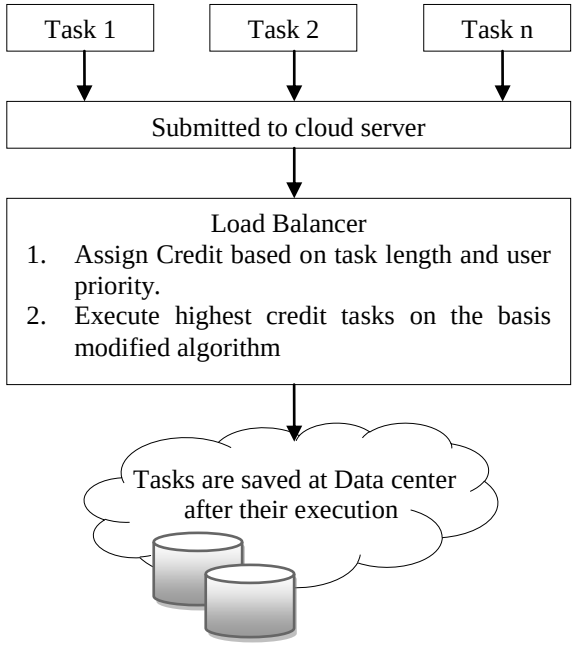


**Figure 2. Proposed Architecture**

The proposed algorithm is based on idea of credit assignment on which each task will be allotted a credit or token and according to assigned credit, tasks are executed so as to improve response time. After assigning credit to each task, each task will be scheduled on the basis of modified algorithm based on round robin load balancing algorithm and throttled load balancing algorithm. The proposed architecture is illustrated as below in figure 2.

## V. CONCLUSION

In this paper, we have presented comparison of different load balancing algorithms for cloud computing such as, round robin (RR), Min-Min, Max-Min, Ant colony, dynamic load balancing, priority based, etc. We described comparative study for these algorithms showing results in different conditions. The vital part of this paper is comparison of different algorithms considering the characteristics like response time, throughput, fault tolerance, overhead, and speed. Future work is to mitigate the

ISSN:2319-6890 (online),2347-5013(print)
1 April. 2017

above problem, and use the hybrid approach to attain better performance and secure the system.

## REFERENCES


i. *K. Li, G. Xu, G. Zhao, Y. Dong, and D. Wang, "Cloud task scheduling based on load balancing ant colony optimization," in Sixth Annu. Chinagrid Conf, pp. 3–9, Aug. 2011.*

ii. *A. Abraham, "Genetic algorithm based schedulers for grid computing systems Javier Carretero, Fatos Xhafa," in International Journal of Innovative Computing, Information and Control, vol. 3, no. 6, pp. 1–19, 2007.*

iii. *M. Katyal and A. Mishra, "A comparative study of load balancing algorithms in cloud computing environment." in International Journal of Distributed and Cloud Computing, Volume 1 Issue 2 December 2013.*

iv. *R. G. Rajan and V. Jeyakrishnan, "A survey on load balancing in cloud computing environments," in International Journal of Advanced Research in Computer and Communication Engineering, vol. 2, no. 12, pp. 4726–4728, 2013.*

v. *X. Xu, H. Yu, and X. Cong, "A qos-constrained resource allocation game in federated cloud," Seventh Int. Conf. Innov. Mob. Internet Serv. Ubiquitous Comput., pp. 268–275, Jul. 2013.*

vi. *B. Godfrey, K. Lakshminarayanan, S. Surana, R. Karp, I. Stoica, "Load balancing in dynamic structured P2P systems", in INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 4, IEEE, 2004.*

vii. *Gaurang Patel, Rutvik Mehta, Upendra Bhoi, "Enhanced Load Balanced Min-Min algorithm for static Meta Task Scheduling in Cloud Computing", ICRTC, Elsevier, 2015.*

viii. *Brototi Mondal, Kousik Dasgupta, Paramartha Dutta, "Load Balancing in Cloud Computing using Stochastic Hill Climbing – A Soft Computing Approach", C3IT, Elsevier,2012.*

ix. *Liang Luo, Wenjun Wu, Dichen Di,Fei Zhang,Yizhou Yan,Yaokuan Mao, "A Resource Scheduling Algorithm of Cloud Computing based on Energy Efficient Optimization Methods", IEEE, 2012.*

x. *Dzmitry Kliazovich, Pascal Bouvry, Samee Ullah Khan, "DENS: Data Center Energy-Efficient Network-Aware Scheduling, Cluster Computing", special issue on Green Networks, vol. 16, no. 1, 2013.*

xi. *Ning Liu, Ziqian Dong, Roberto Rojas-Cessa, "Task Scheduling and Server Provisioning for Energy-Efficient Cloud-Computing Data Centers", IEEE 33rd International Conference on Distributed Computing Systems Workshops Task, 2013.*

xii. *Mrs.S.Selvarani, Dr.G.Sudha Sadhasivam, "Improved Cost-Based Algorithm For Task Scheduling in Cloud Computing", IEEE, 2010.*

xiii. *Xiaonian Wu, Mengqing Deng, Runlian Zhang, Bing Zeng, Shengyuan Zhou, "A Task Scheduling Algorithm based on QoS driven in Cloud Computing", Information Technology and Quantitative management, 2013.*

xiv. *Monir Abdullaha, Mohamed Othmanb, Cost Based Multi QoS Job Scheduling using Divisible Load Theory in Cloud Computing, International Conference on Computational Science, ICCS 2013.*

xv. *Upendra Bhoi, Purvi N. Ramanuj, "Enhanced Max-min Task Scheduling Algorithm in Cloud Computing", International Journal of Application or Innovation in Engineering & Management, Volume 2, Issue 4, April 2013, pages 259-264.*

xvi. *Antony Thomas, Krishnalal G, Jagathy Raj V P, "Credit Based Scheduling Algorithm in Cloud Computing Environment", ICICT 2014,Pages 913-920.*

xvii. *Soumi Ghosh, Chandan Banerjee, "Priority Based Modified Throttled Algorithm in Cloud Computing", IEEE, 2016.*

xviii. *Mr. Shubham Sidana, Ms. Neha Tiwari, Mr. Anurag Gupta, Mr. Inall Singh Kushwaha, "NBST Algorithm: A load balancing algorithm in cloud computing", IEEE, 2016, pp. 1178-1181.*

xix. *Pandaba Pradhan, Prafulla Ku. Behera, B.N.B. Ray, "Modified Round Robin Algorithm for Resource Allocation in Cloud Computing", International Conference on Computational Modeling and Security, ELSEVIER, 2016, pp. 878-890.*

xx. *Hend Gamal El Din Hassan Ali, Imane Aly Saroit, Amira Mohamed Kotb, "Grouped tasks scheduling algorithm based on QoS in cloud computing network", Egyptian Informatics Journal, Elsevier, 2016, pp. 1-9.*