MANTECH Publications

Journal of Research in Computer Science and Engineering
Volume 7 Issue 3
ISSN- 2457-0818 (Online)

A Study on Load Levelling Algorithms in Cloud Computing

Pankaj Kaushal[1], Deepanshu Sharma[2]

Assistant Professor[1], Student[2]

Department of Computer Science Engineering

PKG Group of Institutions (Panipat Chhajpur Campus)

Corresponding Author's E-mail:- *deepanshusharma45@gmail.com[2]*

Abstract

Cloud computing might be a new computer paradigm for accessing and altering internet applications and resources. The major function of this paradigm is resource sharing rather than individual ownership of personal devices or local servers. The phrase "cloud computing" refers to the internet-based computing paradigm. This means that processing power, storage, and software are given to corporate computers in a huge data centre and sent to client devices through the internet. The installation and maintenance expenditures are avoided, which distinguishes this from traditional computer design. When compared to conventional computing, this results in greater resource utilisation and improved customer satisfaction. If not managed properly utilising validated procedures, this might lead to a higher chance of system shutdown or an imbalance in the number of requests or load on a certain system. The goal of this research is to look at some of the existing load-balancing strategies.

Keywords: - *Cloud Computing, Cloud, Computing, Virtual Machines, Load Balancing*

INTRODUCTION

Cloud computing is a new trend emerging in the IT world, with massive infrastructure and resource requirements. The cloud is divided into three fundamental levels: package as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS) (laaS). Cloud computing is an internet-based computing approach in which resources

Page 153-160 © MANTECH PUBLICATIONS 2022. All Rights Reserved

are virtualized and content, or apps, are made dynamic. Users can access these non-physical apps over the web. Private and public are two subsets of the four cloud deployment options.

Cloud computing architectures are both coherently parallel and distributed. This facilitates the processing of several requests at the same time. This is made feasible since the cloud company's pooled resources are scattered but tightly linked. This distribution encompasses the whole globe [7]. Requests may be submitted to any VM from anywhere. This randomization can and does result in uneven work load on various processors, resulting in load imbalance.

One of the most important difficulties in cloud computing is load balancing [7]. Load balancing, as the name implies, is a procedure that uses algorithms to provide an apparent equal distribution of the cumulative burden across all servers and VMs involved. This prevents having a node with no load and another with full capacity. If the request load is appropriately balanced, it may be handled faster, resulting in increased speed and resource utilisation. The LB algorithms are used to ensure that no requests are rejected due to a VM overload. The efficacy of the LB performed in a system is used to evaluate its performance.

Load Balancing

There may be no way to forecast when or how large a request would appear. This increases the danger of a VM being overloaded or under loaded depending on its status when the request was received. Depending on how effective the algorithm is, load distribution or load balancing can help lessen that risk. This influences (saves/reduces) total cost, speed, and, in many ways, system performance.

Goals

- Prevent any complexity-related difficulties for the end user.
- As far as feasible, reduce the challenges that emerge in or during load balancing algorithms.
- Boost system performance by allowing the processors of virtual machines to work at their full capability.
- Maintain the system stability of both the remote and client PCs.
- Manage faults and fall backs wisely to avoid client displeasure in the event of system failure.
- Allow for future changes without requiring extra processing power.

Page 153-160 © MANTECH PUBLICATIONS 2022. All Rights Reserved

Challenges to be Achieved [2]

- **Throughput:** the numerical total of requests previously handled at a given moment. For optimal performance, throughput must be high.

- **Overhead:** The entire amount of processing power required by a request throughout the course of its life there must be very little overhead.

- **Fault Tolerance:** The ability to execute load balancing in the absence of a link, node, or linking failure. A strong fault tolerance is necessary for an algorithm.

- **Migration Time:** The amount of time it takes to transmit (migrate) a certain job between systems or nodes. A system's migration time is anticipated to be short.

- **Response Time:** An LB method reacts after a certain period of time. The time specified is the response time. The reaction time should be as short as possible.

- **Resource Utilization:** The attribute that indicates the total amount of resources used during a load balancing system's operation. The use of resources should be optimised in light of future requirements.

- **Scalability:** An algorithm's capacity to perform efficiently over a wide range of systems with varying amounts of resources and equipment. High scalability is necessary.

- **Performance:** A quantitative measure of all the parameters stated together. The performance of a system or an algorithm must be optimal.

Load Balancing Algorithm

Load balancing [3] techniques seek to optimise the combined load of the VMs. One of the most important procedures is to choose the node(s) with whom the load will be shared. They are classified into two groups: The static LB algorithm is simple to comprehend and implement. It does not refer to the status of the nodes, but it may only be used in particular circumstances. Round Robin is one of the most basic scheduling methods. It employs the division of time notion. The expected completion time for the request is broken into smaller periods. Each VM is allocated a specified interval, and the corresponding divisions are executed by the appropriate nodes. This is referred to as time management. The requests are assigned to

Page 153-160 © MANTECH PUBLICATIONS 2022. All Rights Reserved

VMs based on their statuses as determined by a regularly maintained database or queue (i.e., free or occupied). The queue's order must be rigorously followed, and if the time estimation is erroneous or the provided request segment is not completed, it occupies the last position in the queue and the following request is performed. This method assures that each request is fulfilled with the same power and in almost the same time; in other words, users across the board are not kept waiting for long, regardless of when their request was initiated. [4]

Execution Load: This method again employs the notion of division, but this time it splits the request based on the load and power required, rather than the completion time.

This division and allocation is done ahead of time and is followed until it creates issues or malfunctions. The load balancer employs a data structure based on the first-in-first-out principle (guess what it is) to record the status of various VMs prior to or during the request (in which case the log table is dynamically updated). In this dynamic update, a VM is used if it is idle and capable of performing the present job at hand. [4] The same method is used in the event of an overloaded VM. This is the time when the log database is updated with the new distribution as well as the new overhead or worry nodes. The balancer finds and assigns the work that best meets the criteria for each node at the moment. This approach has yielded encouraging results, but it needs additional processing capacity.

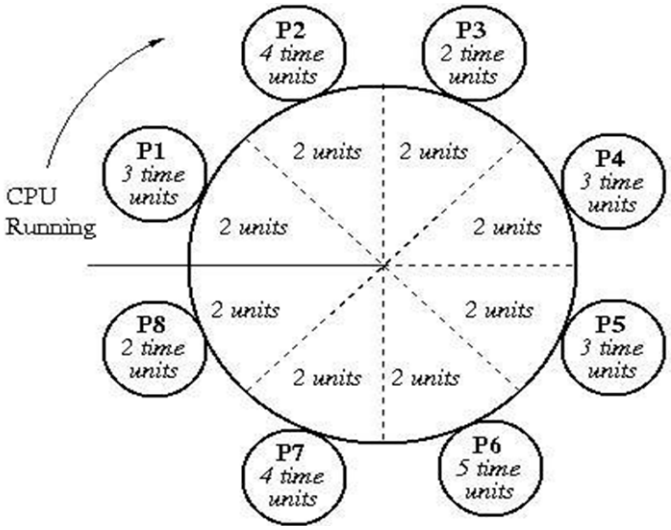


Fig: 1. Depiction of Round Robin

Page 153-160 © MANTECH PUBLICATIONS 2022. All Rights Reserved

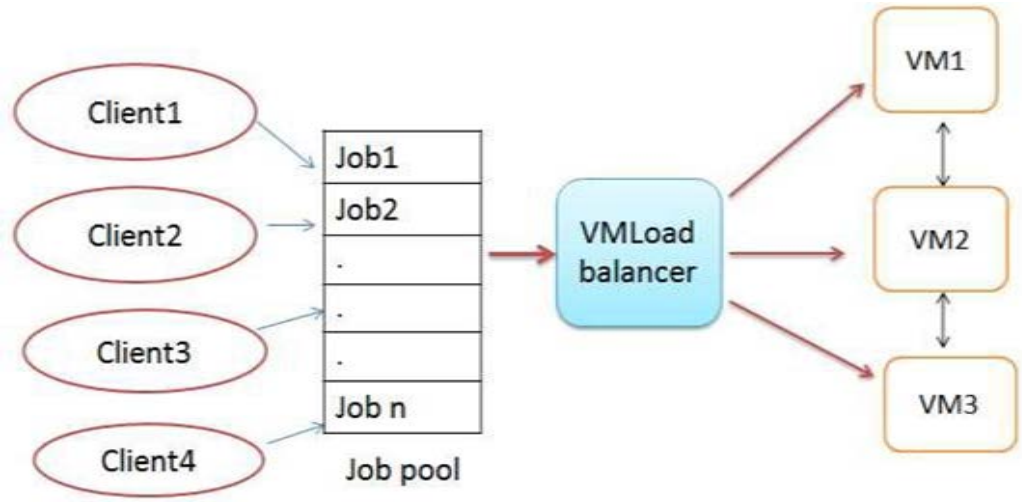


Fig: 2. Depiction of Execution load

FCFS is by far the most straightforward job ordering approach. In this manner, the VMs execute the first request that reaches the VM manager. A FIFO queue can easily handle this.

Throttled: Based on prior request logs, this algorithm receives the request to be executed and determines the time and processing required by the request. Then it searches for a suitable VM or collection of VMs for the given request. If the request is found, it is carried out. Otherwise, it is rejected with a status of -1 for the time being.

Dynamic Algorithms: An LB algorithm that is dynamic in nature and does not take into account the system's past state or behaviour. That is, no log table is kept. The nodes are chosen depending on their existing overhead.

Fuzzy-based Firefly Technique: This algorithm was introduced with the purpose of improving performance by splitting the cloud environment and balancing loads across partitions. Fuzzy logic is used to monitor temporal characteristics during load distribution among virtual machines (VMs). [5]

This logic entails partitioning the entire cluster based on the frequency ordering of the nodes or VMs, i.e., the most frequently used nodes are among the first to be partitioned. These nodes are classified into categories based on the amount of load they can support.

Page 153-160 © MANTECH PUBLICATIONS 2022. All Rights Reserved

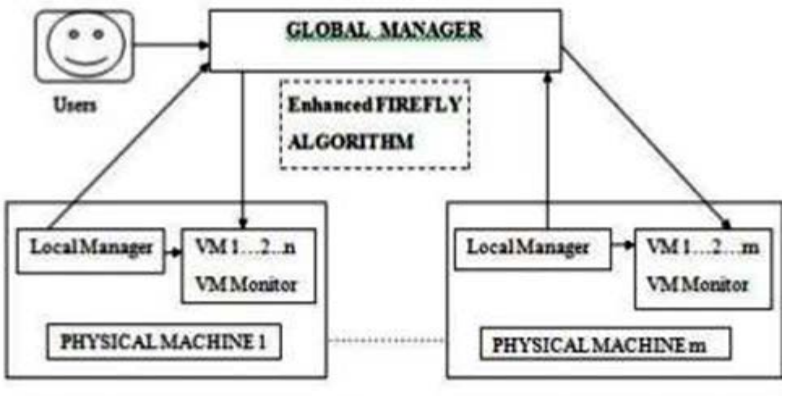


Fig: 3. Firefly algorithm model in cloud computing

Light load, medium load, and high load The load balancer then allocates the requests appropriately. A VM is being compared to a firefly as it tends to “attract itself to light”. The similarity in the characteristics is what is being used here. A Balance Factor is formulated which is given by:[5]

BF = (L * Skb) / (P * e * Scpu) where, L – Length of input file Skb-Size(Kb)

P - Processing capacity VM e– Processingelement

Scpu - Speed (CPU cycles)

If 0 < BF <1, the load balancing method is likely to be executed.

Without a few adjustments and migrations, load balancing execution becomes problematic when BF > 1. As a key "fuzzification" phase, the algorithm provides predetermined values to requests such as "high," "medium," and "low." The "fuzzy engine," or load balancer, chooses the output based on certain specific criteria and a "defuzzification" procedure. The result is calculated using the Smallest of the Maximum (SOM) approach.Following the experimental analysis, it is concluded that the execution time required by this rule is less, the migration and processing value are less, it has the capacity to manage significant numbers, and it may balance the quantity of hundreds arriving. Ant Colony Optimization Based Load Balancing Algorithm: The goal of this rule is to find an optimal path between the supply of food and the hymenopteran colony based on their behavior. Ants make a record of every node they visit and save that information for future use. To do this, they leave pheromones along their path for other ants to choose the next node. The same concept is used here when the starting secretion level might be a positive whole number, x = 0.

Page 153-160 © MANTECH PUBLICATIONS 2022. All Rights Reserved

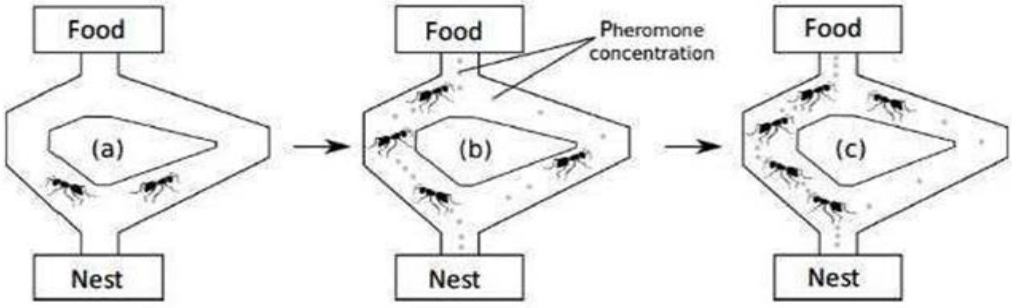


Fig: 4. Ant Colony Concept in Real life [6]

The notion behind VM allocation is that a hymenopter symbolises a VM that looks for a bunch to complete the allocation and so execution. A master database is created that contains the tiny print of each VM. Then, a list of all the hosts that may be assigned to the VMs is created. Then a group is picked on the basis that it has the necessary processing power, memory, and information measures to satisfy the VM's conditions/specifications. Every iteration, data about the masses in a bunch is entered into the database, just like entirely separate ants charting and storing data about each node.When an ant is unable to fulfil a particular task, which suggests that it has not been assigned to a group of VMs, a fall-back approach is implemented. The load information table serves as a secretory path. When a bunch is connected with an ant, the least loaded host from the load information table is picked. When a VM is assigned a job, a planning rule (usually round-robin scheduling) is used. The throughput (explained above) is enlarged as the number of maintained users increases, resulting in high correspondence.

CONCLUSION

This article summarises and describes the many load equalisation strategies that are now or have previously been employed. However, in the realm of cloud computing, a range of extremely inventive and cost-effective load balancing approaches are used. Those algorithms also help future studies, making this a worthwhile analytical topic.

REFERENCES

1. JaspreetKaur “Comparison Load Balancing Algorithms in A Cloud”, International Journal of Engineering Research and Applications, pp. 1169- 1173, 2012

2. Dinesh Babu .L.D, P. Venkata Krishna, Honey Bee inspired load balancing of tasks in cloud

Page 153-160 © MANTECH PUBLICATIONS 2022. All Rights Reserved

computing environment, Applied Soft Computing, 13, Elsevier 2013, 2292-2303.

3. Neethu M.S, "Load balancing in Cloud computing", 2nd International Conference On Recent Innovations In Science, Engineering AndManagement, {22, November, 2015}

4. K. Nishant, P. Sharma, V. Krishna, C. Gupta, K. Singh, N. Nitin, and R. Rastogi, "Load balancing of nodes in cIoud using ant colony optimisation", in Computer Modelling and Simulation (UKSim), 2012 UKSim 14th International Conference on, March 2012, pp. 3{8}

5. J. P. Tushar Desai, A survey of various load balancing techniques and challenges in cloud computing, "International Journal of Scientific & Technology Research", vol. 2, no. 11,pp. ISSN 2277 {8616, nov 2013}

6. Basic concept and terminology of cloud computing- http://whatiscIoud.com

7. S. Mohapatra, K. S. Rekha, and S. Mohanty, "Article: A comparison of four popular heuristics for load balancing of virtual machines in cloud computing", International Journal of Computer Applications, vol. 68, no. 6, pp.33{38, April 2013, published by Foundation of Computer Science, New York, USA.

Page 153-160 © MANTECH PUBLICATIONS 2022. All Rights Reserved