JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Comparison Analysis of Load Balancing Algorithms in Cloud Computing

[1]Shivani Dubey, [2]Mohd. Shabib Raza, [3]Satyam Sharma, [4]Sayeed Ahmed
Department of Information Technology,
Greater Noida Institute of Technology (Engineering Institute), Greater Noida, India

Abstract : Today, cloud computing capabilities are very popular among users. Demand management on cloud platforms faces challenges due to the huge demand from customers for resources and services. A load balancing mechanism is used to solve this problem. In this review, we review, analyze, and identify areas where air load balancing can be applied and how it can be applied. To accomplish this objective, we distribute the product equally among the products according to the system it uses. We also review the performance of the overall application using the equivalent properties and methods of the algorithms. We primarily consider classical load balancing algorithms because they serve as the fundamental algorithms in the load balancing paradigm. Additionally, it pertains to the utilization of pre-existing counterparts within the cloud paradigm.to manage large applications. Researchers can utilize this article to aid in their design process and map the balance of information sources and cloud computing processes.

IndexTerms **- Cloud Computing, virtualization, Load Balancing**

I. INTRODUCTION

The dynamic realm of Cloud Computing empowers users to deploy applications with enhanced scalability, availability, and fault tolerance. It utilizes an internet-based platform for computing technology and encompasses a variety of computing concepts. Cloud computing automatically manages and consolidates computing resources, and it has become an essential tool for storing public and personal information. This technology provides hardware, software, and services according to users' requirements. A cloud computing architecture is distinguished by on-demand self-service, internet accessibility, resource pooling, and service elasticity, and measurement of individual user service utilization. It provides a collection of resources, such as data storage, network, computing power, and specialized applications for both corporate and individual use. Cloud computing offers virtual resources to organizations at a low cost, and its popularity is largely due to its ability to provide virtualization. The resources are provided based on the customers' requirements, making cloud computing highly flexible. While cloud computing provides numerous benefits, it also presents several challenges, including load balancing, task scheduling, VM migration, security, and others that need to be addressed. This paper concentrates on the efficient implementation of load balancing in cloud computing, which has gained increasing popularity and the growing number of intensive tasks waiting to be processed, the fair allocation of cloud tasks becomes critical to ensure that the nodes in the cloud computing environment have a balanced workload. Load balancing plays a significant role in enhancing resource utilization, increasing throughput, decrease the response time and prevent overloading any individual resource. By achieving these goals, load balancing can make cloud computing more efficient and improve user satisfaction. In simple terms, load balancing is a process of distributing work equally among a pool of system nodes or processors to ensure that running tasks are completed without any disruption. Cloud load balancing can be performed individually or on a classified basis and involves the use of various algorithms designed to balance the load with different tasks.

II. CLOUD COMPUTING

In cloud computing, resources are abstracted and presented to users in a virtualized manner, which means that the underlying physical resources are hidden from the users. Cloud computing provides users with the ability to access resources on demand, without requiring users to have any knowledge about the underlying infrastructure. Cluster and Grid computing, on the other hand, do not offer the same level of virtualization and abstraction as cloud computing. In a cluster, a group of computers are connected to work together as a single entity, In contrast to load balancing, Grid computing involves integrating resources from distinct administrative domains to establish a distributed computing environment. Although cloud, cluster, and grid computing share certain characteristics, cloud computing is a different model that offers special advantages including virtualization, scalability, and on-demand resource supply.

Cloud customers or consumers, known as end users, make use of the infrastructure, software, and platform services offered by the cloud. In order to use these services, users are required to accept the terms of the Service Level Agreement (SLA) that the cloud provider has established. End users access services when necessary and are charged based on their usage of the services. Cloud computing incorporates utility computing to provide users with flexibility in accessing its services. Prior to agreeing to the SLA, end users must ensure that it includes specific Quality-of-Service (QoS) parameters that are essential for their needs before utilizing cloud services. Table enumerates some of the fundamental needs or issues that cloud consumers have. Cloud Computing presents end users with a secure and convenient means of accessing any type of infrastructure, software, or platform on an on-demand basis at a reduced cost. Public, private, and hybrid clouds are the three types that cloud service providers can offer. The provider is responsible for constructing the cloud infrastructure.

Enterprises or businesses exclusively own private clouds for internal use. These private clouds can be used to manage and store a lot of data for an organization or to give clients or a team of employee's access to resources whenever they need them. Private clouds are known for providing the highest level of security. Public cloud services include Amazon Web Services, Google Compute Engine, Microsoft Azure, and HP Cloud. A hybrid cloud is formed by integrating both public and private clouds, providing businesses with the ability to manage certain resources within the organization while also utilizing external resources. However, the overall management complexity increases along with security concerns. To optimize the utilization of private and public clouds or their combination, CliQr facilitates businesses in adjusting to changing user needs. Cloud providers are responsible for the task of "resource provisioning," which encompasses two main responsibilities. The first is managing a vast array of resources that constitute the cloud, and the second is providing these resources to end-users.

Cloud developer plays a vital function in the growth of cloud services and applications. They are responsible for designing and implementing the cloud-based solutions that meet the needs of both end users and cloud providers. Cloud developers should have a thorough understanding of cloud architecture and should be able to design scalable and reliable solutions that can handle a large amount of traffic and data. Cloud developers should also ensure that their applications are secure and comply with the relevant regulations and standards. They should be familiar with the programming languages, tools, and frameworks used for developing cloud-based applications, such as Python, Java, Microsoft Azure, and Google Cloud Platform, Amazon Web Services (AWS). Moreover, cloud developers must keep themselves updated with the latest advancements in Cloud technology and be able to adopt new tools and technologies as per the requirement. They should also be able to work collaboratively with other developers, IT professionals, and stakeholders to ensure that the cloud-based solutions are delivered on time and within the budget.

III. SERVICES OF CLOUD COMPUTING

Cloud computing offers multiple clouds to deliver various services, which can vary in size, and usage of these services is measured to determine the corresponding customer billing. The fundamental service models that form the foundation of cloud computing are known as Service Models. These models are typically classified into three primary categories, as outlined below:

a) Infrastructure as a Service

Infrastructure as a Service (IaaS) provides with the basic building blocks of computing infrastructure, including servers, storage, networking, and other resources. Then, clients can employ these resources to create and oversee their own services and apps.

b) Software as a Service

Customers are able to obtain software applications and services via the internet thanks to the cloud computing concept known as Software as a Service (SaaS). This is typically provided on a subscription basis, enabling users of the programme to access it without having to install or maintain it on their own devices. SaaS provides customers with a cost-effective and scalable solution for accessing software and services, as well as the flexibility to adjust their usage based on their needs. **Platform as a Service** Cloud computing uses the Platform as a Service (PaaS) concept where cloud service providers offer a platform for developing, testing, and deploying applications. This includes support for operating systems, execution of programming languages, and other tools and services required to create and manage applications. Customers can focus on their application development without worrying about managing the underlying infrastructure.

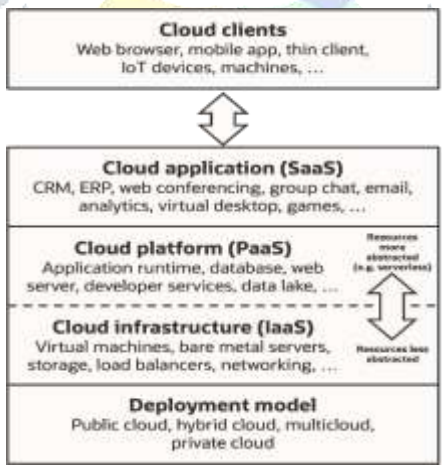

Fig.1: Cloud computing service models arranged as layers in a stack

IV. LOAD BALANCING IN CLOUD COMPUTING

A load balancer is a device, either in software or hardware form, that is used to prevent any one server from becoming overloaded with traffic. The load balancer accomplishes this by distributing network traffic across multiple servers. A load balancing algorithm is a set of predefined rules that the load balancer employs to determine how to distribute the traffic effectively among the servers. To distribute the workload, load balancing is primarily used efficiently across nodes in a network to ensure that no node becomes overburdened with traffic or underutilized. The objective is to maximize the use of available resources while minimizing response times and preventing system failures due to overload. In the Cloud computing environment, several criteria determine the efficacy of load balancing algorithms.

a) **Throughput**: This refers to the total number of executed jobs, and a higher throughput is essential for optimal system performance.

b) **Associated Overhead:** This is referring to the load balancing algorithm's overhead, which needs to be kept to a minimum to be effective. To obtain the best load balancing, the method should be implemented with the least amount of overhead possible.

c) **Fault Tolerance:** The term being described here is the fault tolerance of the algorithm, which is its ability to continue functioning correctly and consistently even in the event of a node failure within the system.

d) Response Time: The response time of a distributed system refers to the minimum duration required for the system to react when a certain load balancing strategy is used.

e) **Resource Utilisation:** The efficiency of a load balancing algorithm may be evaluated by tracking how well system resources are being used. An effective load balancing algorithm should aim to maximise resource consumption.

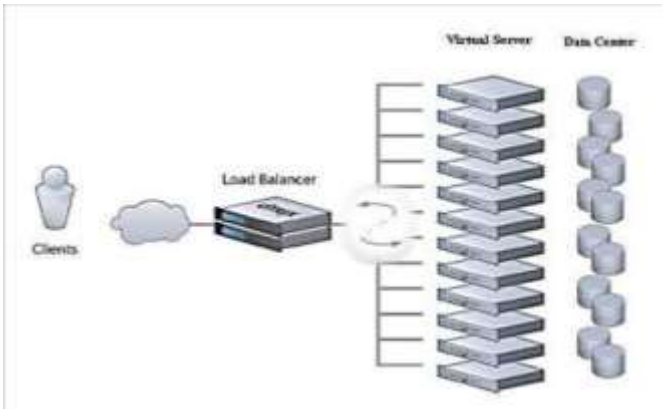

Fig.2: Load Balancing in Cloud Architecture

V. LOAD BALANCING CLASSIFIED AS FOLLOWS ACCORDING TO THE CURRENT STATE OF THE SYSTEM

Algorithms for load balancing can be generally divided into two main categories: static and dynamic algorithms. The various types of load balancing algorithms fall under one of these two categories. In the next section, we will explore these two main groups of load balancing algorithms in detail.

a) Static Algorithms

Static load balancing algorithms are typically only suitable for systems with minimal variations in workload, as they do not take into take into account the system's present status when deciding how to distribute the load. These algorithms require prior knowledge of the system's resources, and at the beginning of execution, Virtual machines' performance is assessed. The burden is subsequently distributed to additional slave processors by the master processor, and the latter receives the output from the slave processors. Because of the non-pre-emption property of static load balancing algorithms, each machine has at least one job assigned to it. The primary goal of these algorithms is to minimize task execution time and limit communication overhead and delays. However, one disadvantage of static load balancing algorithms is that tasks cannot be reassigned during their execution. There are various static load balancing techniques, such as Enhanced Map Reduce, Round Robin, Min-Min, Max-Min, and CLBDM.

b) Dynamic Algorithms

The cloud provider installs diverse and flexible resources in dynamic environments, relying on run-time statistics rather than prior knowledge. In such settings, users' requirements are granted flexibility and can change at any moment. To achieve load balancing in these dynamic environments, Ren proposed the ESWLC technique, which assigns tasks based on node capabilities and the least weight. Similarly, the LBMM algorithm proposed inuses a three-level framework that builds on the OLB algorithm to allocate resources. Because the cloud is scalable and autonomous, dynamic scheduling is a more suitable option than static scheduling. Although dynamic environments are challenging to simulate, they can adapt well to cloud computing environments.

c) Round Robin Algorithm

A static load balancing method is the Round Robin Algorithm approach that doesn't consider preceding states. It assigns jobs to nodes in a circular fashion starting from a random node. This algorithm doesn't require inter-process communication. However, some tasks may get heavily loaded, so a weighted Round Robin algorithm has been proposed where nodes are assigned specific weights based on which they receive requests. If all nodes have equal weights, then the node with the least traffic is selected.

d) Opportunistic Load Balancing Algorithm

Opportunistic Load Balancing Algorithm is another static load balancing algorithm that doesn't take the virtual machine's present workload into account.. It assigns tasks randomly to nodes to keep them busy, but this algorithm doesn't produce optimal results as tasks are executed slowly, and the execution time of the current node isn't considered.

e) Min-Min Load Balancing Algorithm

Static load balancing is accomplished via the Min-Min Load Balancing Algorithm that identifies the execution and completion time of waiting jobs in advance. It assigns jobs to nodes with the minimum execution time, so that tasks are completed in time. However, the algorithm can lead to starvation, as tasks with maximum execution time must wait for a certain period.Just to clarify, the term "starvation" here means that some tasks may never get executed or may have to wait for a very long time to get executed because tasks with shorter execution times are always prioritized.

f) Max-Min Load Balancing Algorithm

It takes a similar approach to the Min-Min algorithm in selecting tasks for execution. However, it differs in that it prioritizes the completion of minimum time jobs before selecting maximum time jobs. This algorithm is static in nature, with all task execution times pre-determined, which helps to ensure optimal task execution.

g) Ant Colony Based Load Balancing Algorithm

It is a technique for dynamic load balancing that effectively distributes workload across nodes. In this algorithm, the head node is chosen as the regional load balancing node, and ants are utilized to assign tasks to nodes. These ants collect information from the cloud node and move towards the next node while checking if it's overloaded or not. If the node is overloaded, the ant moves forward, and if not, it moves backward and replaces the node it found earlier. The solution set

is automatically updated when the ant updates the result for every movement, and once the ant reaches the target node, it commits suicide to prevent backward movement.

h) Honeybee Foraging Load Balancing Algorithm

It is a particular dynamic load balancing algorithm that draws inspiration from honey bee behaviour. The algorithm categorizes honey bees as either "finders" or "reapers," where finders locate the honey source and communicate its quality and quantity through a waggle dance, and reapers collect the honey. In load balancing, the algorithm combines servers into virtual servers with process queues. Upon receiving a request, the algorithm calculates the profit quality similar to how honey bees do in their waggle dance. The server will only stay if the profit is high; otherwise, it will proceed to forage and indicate if the state is loaded, overloaded, underloaded, or balanced. The algorithm maintains a separate queue for each node and removes tasks waiting in overloaded machines, which are then loaded into lightly loaded machines known as scout bees for the next step. While this algorithm reduces response time and waiting time, it does not improve throughput.

i) Biased Random Sampling Load Balancing Algorithm

It is a dynamic approach to load balancing where nodes are represented as servers in a virtual graph constructed based on their connectivity and load. When a request is received by the load balancer, the job is assigned to the node with the minimum in-degree, indicating the availability of free resources. Once the job is completed, the node's in-degree is incremented, indicating the availability of resources. The algorithm uses random sampling to add or remove processes, and a threshold value is used to determine the maximum walk length. If the current node's walk length exceeds the threshold, it moves to another randomly selected node. However, this algorithm's performance decreases as the number of servers increases.

j) Active Clustering load balancing Algorithm

It is an improvement over the Random Sampling method that uses the concept of clustering. This algorithm groups similar nodes together, increasing the efficiency of resource utilization and improving system performance. The algorithm introduces a method called match-maker, where the first node selects a neighbor node and designates it as a Match-Maker node. The match-maker node establishes a connection with the original node's identical neighbour node before cutting the connection. To evenly distribute the load and boost throughput, this process is repeated iteratively. The increased throughput results in the efficient utilization of resources, thus improving system performance.

VI. LOAD BALANCING CAN ALSO BE CLASSIFIED BASED ON THE SPATIAL DISTRIBUTION OF NODES IN THE SYSTEM. THIS CLASSIFICATION INCLUDES THREE TYPES:

a) Centralized Load Balancing: This method of load balancing involves a central server or load balancer that handles the task of distributing the workload among the nodes present in the system. The central server obtains data about the state and usage of each node, and then assigns tasks accordingly. While this technique can be efficient for smaller systems, it may not be suitable for larger distributed systems due to scaling issues.

b) Distributed Load Balancing: This particular load balancing algorithm uses the workload is distributed among multiple load balancers or nodes in the system. Each load balancer is responsible for managing a subset of nodes and distributing tasks among them. This approach is more scalable than centralized load balancing but can be more complex to implement and manage.

c) Hybrid Load Balancing: This approach combines elements of both centralized and distributed load balancing. It uses a central server to manage the overall workload distribution but also includes multiple distributed load balancers to manage subsets of nodes in the system. This approach can provide a good balance between scalability and manageability.

Table.1: Various Load Balancing Algorithms Comparison

Type of Algorithm	Knowledge Base	Issues to be addressed	Usage	Drawbacks
Static	Prior knowledge base is required about each node statistics and user requirements.	Response time Resource utilization Scalability Power consumption and Energy Utilization Makespan Throughput/Performance	Used in homogeneous environment.	Not Flexible Not scalable Is not compatible with changing user requirements as well as load
Dynamic	Run time statistics of each node are monitored to adapt to changing load requirements.	Location of processor to which load is transferred by an overloaded processor. Transfer of task to a remote machine. Information Gathering. Load estimation. Limiting the number of migrations. Throughput	Used in heterogeneous environment.	Complex Time Consuming
Centralized	Single node or server is responsible for maintaining the statistics of entire network and updating it from time to time.	Threshold policies Throughput Failure Intensity Communication between central server and processors in network. Associated Overhead	Useful in small networks with low load.	Not fault tolerant Overloaded central decision making node
Distributed	All the processors in the network responsible for load balancing store their own local database (e.g. MIB) to make efficient balancing decisions.	Selection of processor that take part in load balancing. Migration time Interprocessor communication Information exchange criteria Throughput Fault tolerance	Useful in large and heterogeneous environment.	Algorithm complexity Communication overhead
Hierarchical	Nodes at different levels of hierarchy communicate with the nodes below them to get information about the network performance.	Threshold policies Information exchange criteria Selection of nodes at different levels of network Failure intensity Performance Migration time	Useful in medium or large size network with heterogeneous environment.	Less fault tolerant Complex

VII. CONCLUSION

Using load balancing in cloud computing systems is a crucial process that aims to maximise resource utilisation. This research article examines multiple load balancing strategies, each with its own benefits and drawbacks. Simulate and track static load balancing techniques are rather simple. but they encounter challenges in modeling the varied nature of cloud computing environments. In comparison, Algorithms for dynamic load balancing are better suited for managing heterogeneous environments, but their simulation is more intricate. Furthermore, the effectiveness of the algorithm depends significantly on its implementation level, whether static or dynamic. While distributed algorithms offer better fault tolerance, they require more replication, and hierarchical algorithms distribute the workload at various hierarchical levels. In hierarchical algorithms, upper-level nodes divide tasks equally among the lower-level nodes Dynamic load balancing techniques therefore offer better performance in hierarchical or distributed environments. However, modeling task dependencies with workflows can further boost the efficiency of cloud computing environment.

REFERENCES

1. A.kumar “Load Balancing in Cloud Data Center Using Modified Active Monitoring Load Balancer” IEEE2016.
2. P.Geetha and Dr.C.R.Rene Robin “A Comparative-Study of Load-Cloud Balancing Algorithms in Cloud Environments” International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS-2017).
3. Reena Panwar, Prof. Dr. Bhawna Mallick” Load Balancing in Cloud Computing Using Dynamic Load Management Algorithm”IEEE 2015.
4. Shang-Liang Chen, Yun-Yao Chen ∗, Suang-Hong Kuo “CLB: A novel load balancing architecture and algorithm for cloud services.
5. Surbhi Kapoor, Dr. Chetna Dabas “Cluster Based Load Balancing in Cloud Computing” IEEE 2015.
6. Load Balancing computing [EB/OL], http://en.wikipedia.org.
7. Hamid Shoja, Hossein Nahid “A Comparative Survey On Load Balancing Algorithm In Cloud Computing” IEEE 33044.
8. Radojevic, B &Zagar.M, “Analysis of issues with load balancing algorithms in cloud environment proceedings of 34th International Convention on MIPRO, IEEE .
9. Anju Baby J. A survey on honey bee inspired load balancing of tasks in cloud computing. International Journal of Engineering Research and Technology. 2013 Dec 18; 2(12):1442-5.
10. Singh A, Juneja D, Malhotra M. Autonomous agent based load balancing algorithm in cloud computing. Procedia Computer Science. 2015 Dec 31; 45:832-41.
11.Open Stack: An Overview. Retrieved from www. openstack.org/downloads/openstack-overview-datasheet.pdf.