Classification of Distributed Load Balancing in Cloud Computing

Ms. Vatsala Johari[1], Dr. Rohitashwa Pandey[2]

[1]Research Scholar M. Tech-CSE, Department of Computer Science and Engineering, Bansal Institute of Engineering and Technology, Lucknow.
[2]Associate Professor, Department of Computer Science and Engineering, Bansal Institute of Engineering and Technology, Lucknow.

Abstract:
Distributed load balancing is most important across many servers and allocate the resource allocation and compute the resources in cloud computing. The workload could be distributed in between two or more servers, hard disks, networking devices, or any computing resource. This improves the system's response time and utilization. From the load balancing , servers performance is improved and running smoothly and also maintenance of system speed, stability, and prevention from system breakdowns. Cloud load balancing has the potential to provide scalability for traffic control on a portal. It is feasible to control peak traffic utilizing effective load balancers, which is accomplished with networking devices and servers. The major purpose for using a load balancer is to prevent the website or an online mobile application of any unexpected problems. he cost of cloud load-balancers is depending on the quantity of resource consumed, which is known as the 'pay-as-you-go' paradigm.

Keywords: Distributed Load Balancing, Cloud Computing, Virtual Machine

Introduction:

In a cloud scenario, a data centre is a collection of various physical parts like compute nodes, storage components, routers, network switches, and firewalls which serve as a main storage location for all types of IT equipment. The cloud service provider routes every user's query for cloud computation to the nearest data centre. The Service Level Agreement monitor evaluates the given request to establish the Quality-of-Service parameters. It also keeps track of the status of received requests.

These issues have been addressed using networked physical servers having distinct identities. Every physical server can host multiple virtual machines that employ a hypervisor to share the physical servers' computational resources. Whenever different resources are available to contest depending on deadlines, incoming traffic is forwarded to such virtual machines inside the server for further processing taking the help of a dispatcher. Such resources are freed and accessible for the development of new virtual machines to serve new client request after the task is completed. Task scheduling is a demanding research subject in cloud computing because tasks must be assigned to the proper resource to achieve optimal performance. As a result, researchers and engineers propose a variety of ways for overcoming the below mentioned challenges while scheduling tasks [1].

- Since cloud computing is based on resource sharing, multiple users contend for the environment's resources.

- Since the resources in the cloud environment could be diverse, they may not behave identically for the task at hand.
- Since a task scheduler breaks down an operation into several activities, there may be relationships between the order in which tasks are executed and how data is communicated.

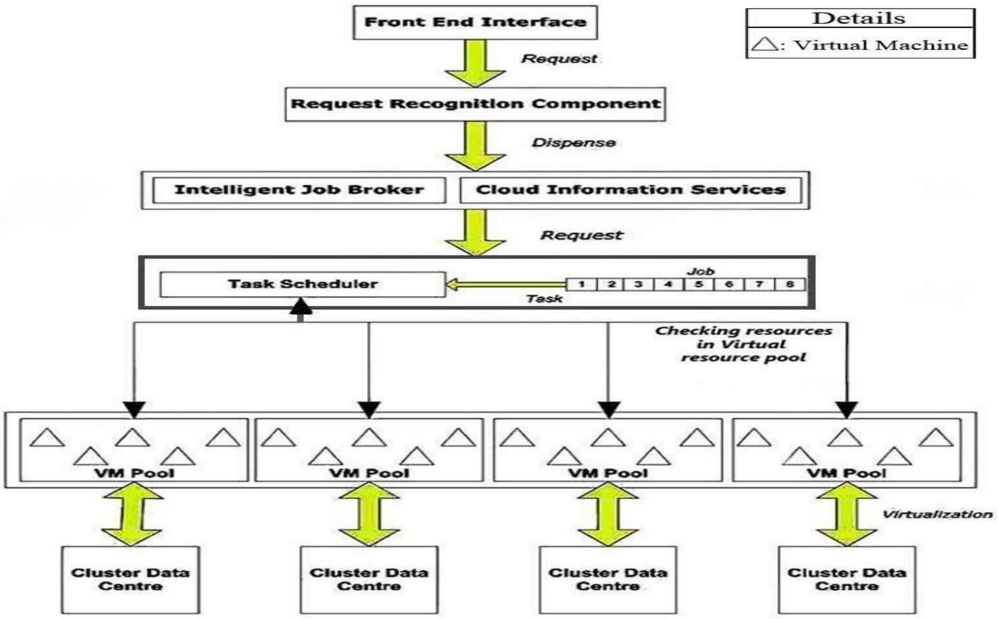

Figure 1.1 General model of task scheduling

Above Fig 1.1 represents the general model of task scheduling in cloud computing environment. It is made up of numerous parts. These task schedulers break down the request from the user into smaller tasks. Such task units are sent to the cloud environment's task queues for further execution. Numerous resources get underlined once the cloud service provider completes the resource discovery and tracking stage of managing resources. The virtual machine supervisor has access to all operational virtual machines, as well as the length of each server's local task queue and resource requirements. In a cloud environment, the virtual machine manager analyses availability of resources for specific task units. If there are enough operational virtual machines to complete the tasks, the manager passes them to the task scheduler; otherwise, new virtual machines are built on servers with available resources.
Energy management is aided by resource optimization, that is tracked by the energy monitor.[5]

Load Balancing Architecture:

Load balancing is performed at two levels. The physical machine or host level is the first layer in load balancing, and the virtual machine level is the second stage in load balancing. A virtual machine manager and monitor are used to carry out this mechanism. These two components are abstracted in the load balancing architecture as shown in fig 2.1 To balance the load task can be migrated at the physical machine level and the virtual machine level and can be categorized as Intra VM task migration and Inter VM task migration.

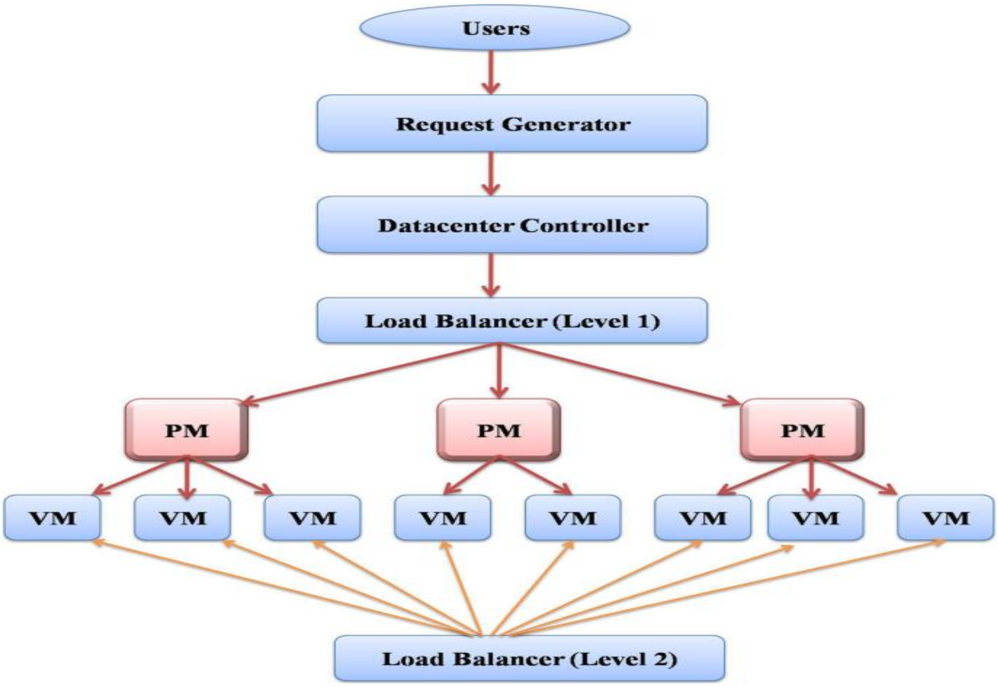

Figure 2.1 General Architecture of Load Balancing

In a cloud environment, a consumer's request is generated by the request generator for processing that needs computing resources for the execution. Load balancer checks for the most appropriate virtual machine for the incoming task. At the first level of load balancing, the load is distributed within the single physical machine by distributing the workload among associated virtual machines within the single physical machine. To balance the load, tasks are migrated within the single physical machine that is known as intra task migration. In the second level of load balancing workload is migrated to different virtual machines of the entire system. This is known as inter task migration.[2]

Types of Load Balancing :

Load balancers used to be hardware in private datacenters, but they are transformed into light-weight software application packages in past years. Software-based load balancers usually have virtual machines for processing requests and include extra features like safety, speed, and verification. They can satisfy the mobility and affordability needs of modern businesses seeking a competitive advantage.

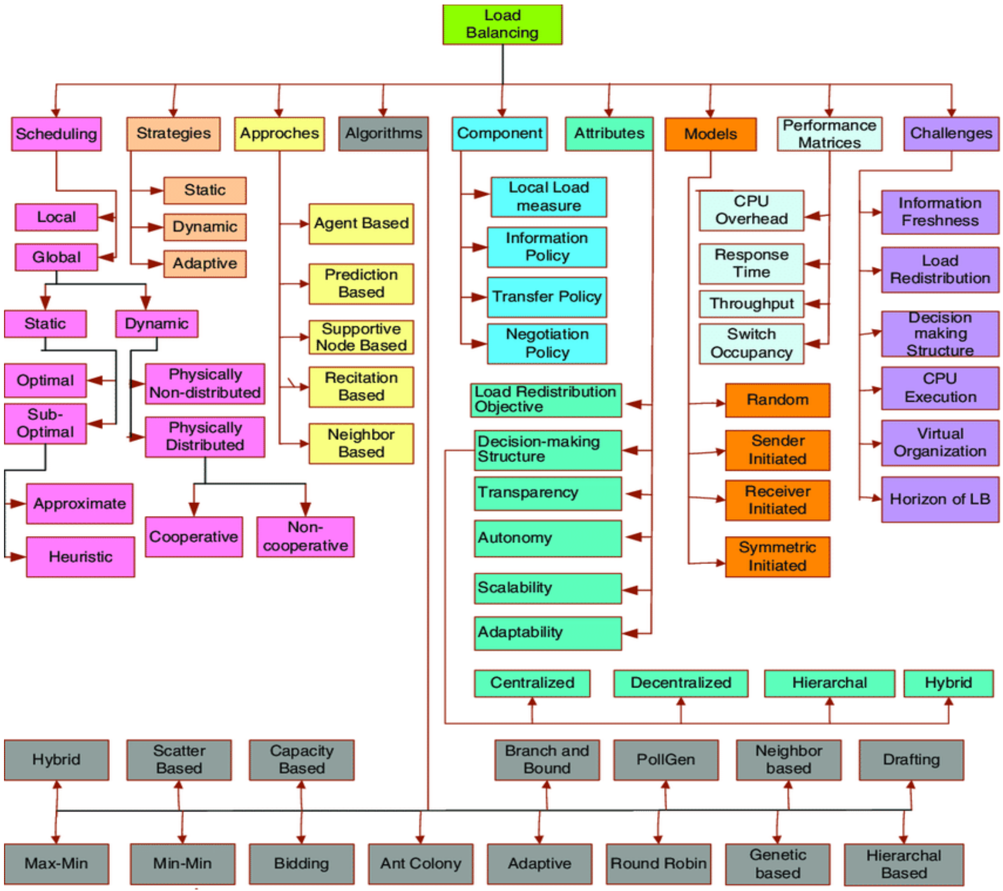

Figure 2.2 Different Classifications of Load Balancing Algorithms

A review of existing load balancing survey articles demonstrates that every study aims to classify parameters in its own way. The above figure represents the different ways of classifying the load balancing algorithms by different authors. We divided the load balancing algorithms based to statistics or inspired by nature. They used the flowcharts to compare the algorithms and represented the algorithms graphically. The algorithms inspired by nature are further classified based on the swarm-based behavior which includes the ant colony optimization, Artificial Bee and Particle Swarm Optimization and based on the evolution by genetic algorithms. when applied to the load balancing algorithms improve the response time and efficiency a lot. The division on the basis of statistics is based on the awareness of the resource and on the basis of performance. The Bin packing, Agent based and dynamic cluster are the subdivisions on the basis of resource while adaptive and QOS based approach is used on the basis of performance.

In this paper, We gave a different classification of load balancing algorithms depending on the state of the system and on the basis of owner of the initiation of the process. According to system state the algorithms can be static or dynamic and a dynamic algorithm can be distributed or non-distributed. Further a distributed algorithm is classified as cooperative or non-cooperative and a non- distributed is classified as centralized or semi- distributed. The static algorithm is based on the prior knowledge of the properties of nodes and attributes of the processes. The static algorithm is limited by the fact that it does not adopt the dynamic changes in the attributes of the system. The algorithm based on the dynamic approach need not acquire any prior knowledge about the system. This approach is more flexible and suitable for heterogeneous environments. In cooperative scheduling distributed entities cooperate to make scheduling decisions to achieve a common goal while in non-cooperative scheduling, tasks are independent and make scheduling decisions independently and also known as independent scheduling. Cooperative algorithms are more complex and involves a larger overhead than noncooperative scheduling. We directly classify the algorithms as static and dynamic. A dynamic distributed algorithm is referred as cooperative or noncooperative algorithms and a dynamic non distributed approach is referred as semi- distributed and centralized. Load balancing decisions based on the centralized approach rely on a single node having all the system-related information and act as a centralized server unit responsible for making scheduling related decisions. In this approach, each node periodically sends messages to the main server node to update the information, and then it can efficiently make decisions about scheduling of processes because it has all the information about the load at a node and the number of processes that need service. While in the distributed approach, process assignment decisions are made by the various nodes of the system which are physically distributed. Each node is responsible for maintaining its state information vector about the other nodes. The distributed approach is more reliable than centralized but increases the network traffic because of more communication between the nodes. In semi- distributed approach, nodes are split into clusters to perform load balancing. We divided the load balancing algorithms according to system load and system topology. Further according to system load, they are sub- classified as centralized, distributed or mixed algorithms. Based on the system topology further subdivision is done as static, dynamic or adaptive algorithms. The differences between resource allocation strategies and task scheduling approaches. These algorithms are classified on the basis of distribution of nodes, environment of cloud and dependencies among tasks. The nodes distribution is centralized, distributed or hierarchical and the cloud environment is considered as static or dynamic while the task dependencies are shown by the workflow-based algorithms. In the hierarchical approach, scheduling of tasks and load balancing is carried out at different levels. It is based on the tree data structure and is a hybrid approach that uses the characteristics of a centralized and distributed approach and is used to lower the communication delay to increase the performance. The formation of nodes is in the form of master and slave and only the master node is responsible fortaking the load balancing decision. Mainly classify the algorithms as static and dynamic. Some examples of static algorithms are given as Opportunistic Load Balancing, Minimum Execution time, Minimum completion time, genetic algorithms, tabu search etc. The dynamic approach is further sub-classified as online or offline mode. While survey papers apparently show different ways of categorizing load balancing heuristics, in synthesis there are only three broad aspects on which these categorizations are made, viz. (i) scheduler set up, (ii) task considerations, and (iii) type of heuristic.

Scheduler set up refers to the arrangement of schedulers. Scheduling can be done by multiple schedulers, referred to as distributed, or by a single scheduler, referred to as centralized, or by a hybrid of the two, called semi-distributed. Centralized design is easy to implement, however, its obvious disadvantage is that there is a single point of failure and thus it lacks fault tolerance. While this can be overcome by using a distributed set of schedulers, the distributed design introduces an additional layer of complexity to the system. Task considerations refers to how task arrival is handled. If tasks are scheduled as soon as they arrive, the scheduling is referred to as online or dynamic, or if tasks are aggregated into groups and scheduled together, it is referred to as batch mode or static. The type of heuristic refers to the underlying concept of the heuristic itself, e.g., it could be a fixed strategy or set of rules such as round robin, or a statistical or nature inspired optimization method.

Challenges:

In terms of conceptual and applied factors, this technology has become the attraction for higher research in the area of data and computing. Yet, cloud computing research is beset by problems, with load balancing one of them that requires careful planning. Following is a list of just few load-balancing challenges :

Geographically dispersed nodes: For computing reasons, data centers with in cloud are generally scattered geographically. In these centers, spatially distributed nodes are treated as a single location system for the seamless execution of client requests. Few load balancing solutions are designed for a smaller area and ignore factors such as network congestion, transmission delay, distance between dispersed processing units, and user-resource segregation. Developing load-balancing methods for nodes that are far apart should be considered.

Failure at a single point: Various dynamic load balancing methods were proposed, some of which are non-distributed and rely on the central node to make load balancing choices. A failure of the central node, on the other hand, will affect the overall runtime environment. As a consequence, a number of distributed algorithms should be designed to ensure that a single node somehow doesn't gain control of the entire computing system.

Relocation of Virtual Machines: Virtualization enables the development of multiple virtual machines (VMs) on a particular physical computer. They are self-contained and have varied capabilities. If a physical system becomes overburdened, some virtual machines should move to a different site by only a balancing mechanism known as Virtual Machine migration.

Diverse nodes: User requirements constantly change in cloud computing, necessitating its execution across heterogeneous nodes to maximise resource utilisation and minimise response time. As a consequence, dealing with diverse nodes is a challenge for researchers.

Administration of storage: Data storage in cloud has solved the problem of traditional file systems that required personnel monitoring and had high equipment costs. Consumers can store a variety of information in the cloud without fear of accessibility. Online storage is rapidly expanding, demanding data redundancy storage for effective data access and stability. Due to the identical data storage policy at replicating places, complete data replication techniques are inefficient. Partial replication may be adequate, but dataset accessibility may be a concern, adding to the complexity of load-balancing strategies.

Algorithm Difficulty: Techniques within cloud computing should be simple and easy to implement. A complicated algorithm would degrade the efficiency and durability of the cloud system.

Conclusion

Most the modern web applications and online mobile applications need the ability to support a large number of online users with ability to scale up or down as the user workload changes and load balancing is a key technique to distribute these workloads across multiple servers and is thus a key research area in Cloud computing. We have discussed in this paper the two aspects of load balancing, one is with regards to scaling physical servers to multiple virtual machines through hypervisors and the other is redirecting of network traffic to multiple virtual machines, via a dispatcher, to balance incoming user requests. Together these two constitute a load balancing architectural framework. Effectiveness of load balancing in cloud computing can be measured through multiple metrics like resource utilization, makespan, cost, energy consumption etc., however Average Response Time and Throughput remain the key ones as they are strongly related to user experience. We also covered the various load balancing classifications found in the current research literature, static vs dynamic centralized vs distributed, nature inspired, statistics based and sub classifications as well like online vs offline mode for dynamic algorithms. As part of this research, we also concluded that across all these classifications there are three key underlying factors on which these categorizations are made, scheduler set up, task considerations, and type of heuristic.

Reference:

1. Zhang, Shufen, Hongcan Yan, and Xuebin Chen. "Research on key technologies of cloud computing." Physics Procedia 33 (2012): 1791-1797.
2. Padhy, Rabi Prasad, and Manas Ranjan Patra. "Evolution of cloud computing and enabling technologies." International Journal of Cloud Computing and Services Science 1.4 (2012): 182.
3. Singh, Manjeet. "An overview of grid computing." 2019 International Conference on computing, Communication, and Intelligent Systems (ICCCIS). IEEE, 2019.
4. Mahmood, Zaigham. "Cloud Computing Technologies for Connected DigitalGovernment." Web 2.0 and Cloud Technologies for Implementing Connected Government. IGI Global, 2021. 19-35.
5. Armbrust, Michael, et al. "A view of cloud computing." Communications of the ACM 53.4 (2010): 50-58.
6. Mell, Peter, and Tim Grance. "The NIST definition of cloud computing." (2011).
7. Gill, Sukhpal Singh, and Rajkumar Buyya. "A taxonomy and future directions for sustainable cloud computing: 360 degree view." ACM Computing Surveys (CSUR) 51.5 (2018): 1-33.
8. Stieninger, Mark, and Dietmar Nedbal. "Characteristics of cloud computing in the business context: A systematic literature review." Global Journal of Flexible Systems Management 15.1 (2014): 59-68.
9. Tadapaneni, Narendra Rao. "Different Types of Cloud Service Models." (2017)
10. Diaby, Tinankoria, and Babak Bashari Rad. "Cloud computing: a review of the concepts and deployment models." International Journal of Information Technology and Computer Science 9.6 (2017): 50-58.