IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 3 Issue 10, October 2016
ISSN (Online) 2348 – 7968 | Impact Factor (2015) - 4.332
www.ijiset.com

Comparative Study on Load Balancing Techniques in Cloud Computing

S.Saranya [1], Dr.R.Manicka Chezian[2]

Research Scholar, Department of Computer Science, Nallamuthu Gounder Mahalingam College,
Pollachi, India[1]
Associate Professor, Department of Computer Science, Nallamuthu Gounder Mahalingam College,
Pollachi, India[2]

Abstract: load balancing is often to provide repetition in the applications. If one server in a cluster of servers fails, the load balancer can temporal remove that server from the cluster, and partition the load onto the functioning servers. Multiple servers help each other in this way is called "redundancy". Whenever an error happens and the task is to move from the failing server to a functioning server, this is generally called "failover". A set of servers running the same application in cooperation is commonly referred to as a "cluster" of servers. Load balancing is one of the cloud environment based on how cloud developers demanded for the cloud provider. This paper explains load balancing and its types, particularly static and dynamic load balancing and performance parameter also compared.

Keywords: Load balancing, load balancing platform and environment, Need of load balancing, Classification of load balancing, performance parameter load balancing.

I.INTRODUCTION

Cloud computing system can be divided into two sections as front end and back end. They both are connected with each other through a network, usually the internet. Front end is what the client (user) sees whereas the back end is the cloud system. Front end has the client's computer and the application required to access the cloud (Browser) and the back has the cloud computing services like on-demand computing and data storage from various servers. The difference between traditional system and cloud system is represented in the next diagram. Using hypervisor, also called virtual machine manager (VMM), is one of many hardware virtualization techniques allowing multiple operating systems, termed guests, to run concurrently on a host computer. It is so named because it is conceptually one level higher than a supervisory program. They both are connected with each other through a network, usually the internet. Front end is what the client (user) sees whereas the back end is the cloud system. Front end has the client's computer and the application required to access the cloud (Browser) and the back has the cloud computing services like on-demand computing and data storage from various servers. The difference between traditional system and cloud system is represented in the next diagram. Using hypervisor, also called virtual machine manager (VMM), is one of many hardware virtualization techniques allowing multiple operating systems, termed guests, to run concurrently on a host computer. It is so named because it is conceptually one level higher than supervisory program hardware, with the function of running guest operating systems, that themselves act as servers. [1][2]

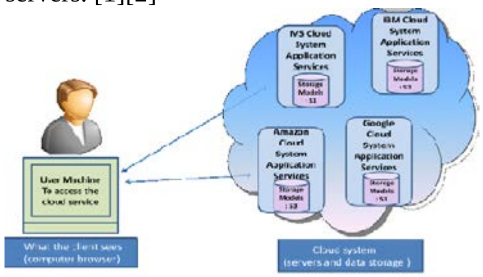

IVS Cloud System Application Services
Storage Models :S1
IBM Cloud System Application Services
Storage Models :S3
Amazon Cloud System Application Services
Storage Models :S3
Google Cloud System Application Services
Storage Models :S1
User Machine To access the cloud service
What the client sees (computer browser)
Cloud system (servers and data storage)

Figure 1:Cloud computing system architecture

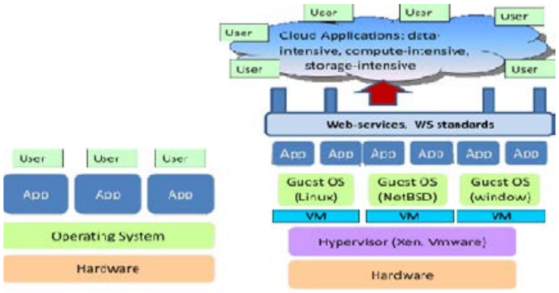


Figure 2: Compare b/w traditional system and cloud system

The figure 1 describes computer system architecture of cloud will process for a user machine through access of the cloud service and level of the cloud storage. The figure 2 describe for the traditional system and cloud system of the cloud environments.

II.LITERATURE REVIEW

D. Zhang ET al.proposed a binary tree structure that is used to partition the simulation region into sub-domains. Dhinesh et al. proposed an algorithm named honeybee behavior inspired load balancing algorithm. Here in this session well load balance across the virtual machines for maximizing the throughput. B. Dong et al. proposed a dynamic file migration load balancing algorithm based on distributed architecture. Considered the large file system there were various problems like dynamic file migration, algorithm based only on centralized system etc. The J. Hu et al. proposed a scheduling strategy on load balancing of VM resources that uses historical data and current

State of the system. Proposed strategy achieves the best load balancing and reduced dynamic migration by using a genetic algorithm. The A. Bhadani et al. proposed a Central Load Balancing Policy for Virtual Machines (CLBVM) that balances the load evenly in a distributed virtual machine/cloud computing environment. The work done by M. Randles et al.investigated a distributed and scalable load balancing approach that uses random sampling of the system domain to achieve self-organization thus balancing the load across all nodes of the system.

III.LOAD BALANCING IN CLOUD PLATFORM AND ENVIRONMENT

Load balancing is a process of reassigning the total load to the individual nodes of the collective system to make resource utilization effective and to improve the response time of the job, simultaneously removing a condition in which some of the nodes are over loaded while some others are under loaded. A load balancing protocol is dynamic in nature doesn't contemplate the previous state or behavior of the system, that is, it depends on the current behavior of the system. It is common these days in redundant high-availability computer systems that incoming network traffic is distributed on network level by deploying one of the frequently used network load balancing These algorithms use solely network parameters of incoming traffic to create selections wherever to forward traffic, with none data from different elements of database system, like current load of application servers.

Cloud computing can have either static or dynamic environment based upon how developer configures the cloud demanded by the cloud provider.

A) Sender Initiated

In this type of load balancing algorithm the client sends request until a receiver is assigned to him to receive his

Workload i.e. the sender initiates the process.

B) Receiver Initiated

In this type of load balancing algorithm the receiver sends a request to acknowledge a sender who is ready to share the workload i.e. the receiver initiates the process.

C) Symmetric

It is a combination of both sender and receiver initiated type of load balancing algorithm. Based on the current state of the system there are two other types of load balancing algorithms.

D) Static Environment

This approach is generally defined in the design or implementation of the system. This algorithm has a drawback that the task is assigned to the processors or machines only after it is created and that task cannot be shifted during its execution to any other machine for balancing the load.

E) Dynamic Environment

This approach takes into account the current state of the system during load balancing decisions. This approach is

More suitable for widely distributed systems such as cloud computing. An important advantage of this approach is that its decision for balancing the load is based on the current state of the system which helps in improving the overall performance of the system by migrating the load dynamically. [4][6]

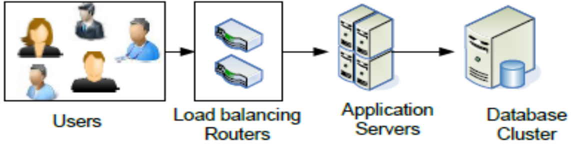


Figure 3: computer System with hardware load balancers

The figure 3 describes computer system hardware of load balancers will be working for the users, load balancing routers, application servers and database cluster.

V.NEED OF LOAD BALANCING

We can balance the load of a machine by dynamically shifting the workload local to the machine to remote nodes or machines which are less utilized. This maximizes the user satisfaction, minimizing response time, increasing resource utilization, reducing the number of job rejections and raising the performance ratio of the system. Load balancing is also needed for achieving Green computing in clouds. The factors responsible for it is: Limited Energy Consumption: Load balancing can reduce the amount of energy consumption by avoiding over hearting of nodes or virtual machines due to excessive workload. Reducing Carbon Emission: Energy consumption and carbon emission are the two sides of the same coin. Both are directly proportional to each other.

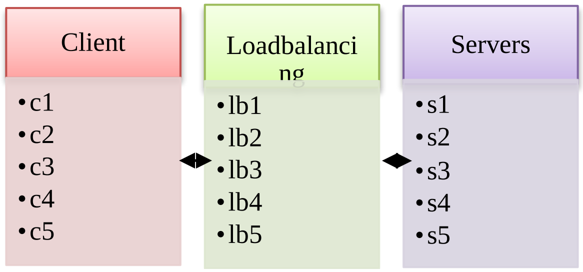


Figure 4: Load Balancing in Process

The figure 4 describes Load balancing is a process of reassigning the total load to the individual nodes of the collective system to make resource utilization effective and to improve the response time of the job. Simultaneously removing a condition in which some of the nodes are over loaded while some others are idle.

VI.CLASSIFICATION OF LOAD BALANCING

Broadly, load balancing is a kind of scheduling optimization problem. The load balancing strategy may be determined by inspection, such as with a rectangular lattice of grid points split into smaller rectangles, so that the load balancing problem is solved before the program is written. Depending on the information used in load balancing decision, it can be divided into two broad categories global and local policies. In global policies, the load balancer uses the performance profiles of all available workstations. In local policies workstations are partitioned into different groups. Static load balancing algorithms rely on the estimate execution times of the tasks and inter-process communication requirement. It is not satisfactory for parallel programs that are of the dynamic and/or unpredictable kind. Consequently in dynamic load balancing, tasks are generated and destroyed without a pattern at run time.

Further, depending on the location where the load balancing decision is carried out the resident of the load balancer, these can be further classified either as centralized or distributed load balancing. The case when the load balancer resides at the master node is called centralized load balancing policy, otherwise if the same resides at all the workstations under consideration is called the distributed load balancing policy. Because the change is discrete, the load balance problem and hence its solution remain the same until the next change. If these changes are infrequent enough, any savings made in the subsequent computation make up for the time spent solving the load balancing problem. The difference between this and the static case is that the load balancing must be carried out in parallel to prevent a sequential bottleneck. [7]

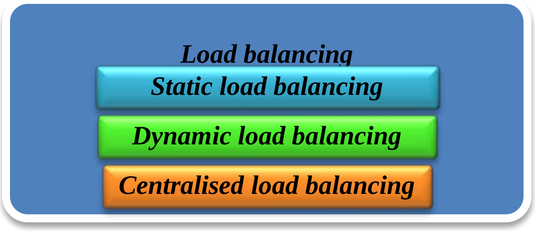


Figure 5: various types of classification load balancing

The figures 5 describes classification of load balancing global part will be two parts one is static load balancing and another dynamic load balancing some other types of load balancing.

STATIC LOAD BALANCING

Static load balancing is based on the concept of master and slave. The performance of processor/server nodes is determined at the beginning of execution and on the basis of that performance workload is assigned by master node. The slave nodes evaluate their work and submit result to master node. The tasks are assigned to slave nodes at compile time, i.e., no change or reassignment is possible at runtime. There is little overhead involved when using static load balancing as there is no migration of jobs at runtime. The tasks are assigned randomly, in a cyclic fashion or based on size range.

A) Random Allocation: Tasks are assigned randomly to the servers. This method uses random numbers to select processors which are generated on the basis of some statistical distribution. Among N servers, server I receives the task with probability 1/N.

B) Round Robin Allocation: The tasks to the server are allocated in a cyclic fashion. Once the server is assigned a task it is moved to the end of the list so that load is equally shared. Random allocation and round robin policies are simple to implement but their performance deteriorate when the task size distribution is heavy tailed.

C) Threshold Based Allocation: The processes are allocated to the server nodes as soon as they are created. Each server node has a private copy of system's current load. The current load of server node could be under loaded, medium loaded and over loaded. Threshold parameters describe these levels:
Load < thunder under loaded
t_under <= load <= t_upper medium
Load > t_upper overloaded

D) Central Manager Allocation: The central processor/node selects the host for new process which has least node using the server's current system load information. All the remote server nodes update the load information by sending a message every time when their load changes.

E) Size Based Policies: In these policies, size ranges are affected to the servers. The dispatcher assigns the tasks on the basis of size. *SITA-E* (Size Interval Task Assignment with equal load): The main idea behind this approach is that workload assigned to each cluster will be equal. SITA-E performs poorly when variability increases. *SITA-V* (Size Interval Task Assignment with variable load): When the tasks arrive, the dispatcher assigns all the small tasks to the server that is under loaded and the large tasks to the server that is over loaded. The main idea is to reduce the mean slowdown. *SITA-U* (Size Interval Task Assignment with unbalanced load): This policy uses the concept of cut-off to assign tasks to the server. Each server has a cut-off range and accepts only those tasks that fill in its cut-off.

DYNAMIC LOAD BALANCING

Unlike static load balancing policies, the dynamic policies attempt to balance load based on some load information at the servers. Job/task assignment is done at run time. It results in better decision making as compared to static load balancing but load balancer has to monitor the current load on all nodes continuously, it becomes an extra overhead as monitoring consumes CPU cycles. Moreover, dynamic schemes spend more time in migration of jobs than executing any useful work.

A) Central queue algorithm: All new activities and unfulfilled requests are stored as cyclic FIFO queue on the main host. Each new activity which arrives is inserted into the queue. When the request for the activity is received it is removed from the queue.
Local queue algorithm: The basic idea behind this allocation is static allocation of new processes with process migration initiated by host when its load falls under threshold limit which is user defined parameter that defines the minimal number of ready processes the load manager attempts to provide on each processor.

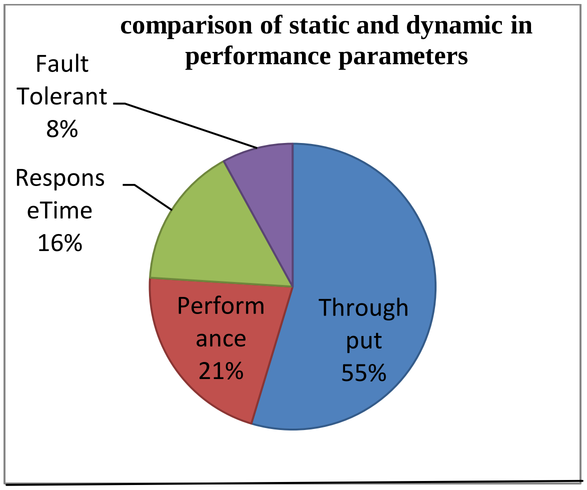
B) Least Loaded First (LLF): Tasks are assigned to the server on the basis of its load. Examples of least loaded first mechanism are shortest queue and least work remaining. In shortest queue, based on the queue's contents the dispatcher assigns the incoming tasks so that the server that has least number of tasks in the queue will be assigned the next task for processing. Least work remaining uses the remaining work at the server to dispatch the tasks and then selects the server that has least of the remaining work in terms of task size.

C) Cycle stealing with central queue: This is a variant of the central queue policy. The servers are divided into two different groups: the first one is the group of short servers which processes the small tasks and the second is the group of long servers to process the large tasks. When one of the short servers becomes free, it picks the small task in the queue for processing. Similarly if one of the long servers becomes free it processes large task. If there is no large task long server processes small tasks and when there is no small task small server processes long tasks. Now we will compare static load balancing and dynamic load balancing considering few parameters. [5][3]

VII.PERFORMANCE MATRIX FOR LOAD BALANCING

The different qualitative metrics or parameters that are considered important for load balancing in cloud computing are discussed as follows:
1. Throughput: The total number of tasks that have completed execution is called throughput. A high throughput is required for better performance of the system.
2. Associated Overhead: The amount of overhead that is produced by the execution of the load balancing algorithm. Minimum overhead is expected for successful implementation of the algorithm.
3. Fault tolerant: It is the ability of the algorithm to perform correctly and uniformly even in conditions of failure at any arbitrary node in the system.
4. Migration time: The time taken in migration or transfer of a task from one machine to any other machine in the system. This time should be minimum for improving the performance of the system.

5. Response time: It is the minimum time that a distributed system executing a specific load balancing algorithm takes to respond.
6. Resource Utilization: It is the degree to which the resources of the system are utilized. A good load balancing algorithm provides maximum resource utilization.
7. Scalability: It determines the ability of the system to accomplish load balancing algorithm with a restricted number of processors or machines.
8. Performance: It represents the effectiveness of the system after performing load balancing. If all the above parameters are satisfied optimally then it will highly improve the performance of the system. [8]



Parameters	Static	Dynamic
Throughput	Low	High
Associate overhead	Low	High
Fault tolerant	Low	High
Migration time	High	Low
Response time	Low	High
Resource utilization	Low	High
Scalability	Low	High
Performance	Low	High

The figure 6: comparison of static and dynamic load balancing and Table 1: compare load balancing techniques

The Figure 6 throughput and performance parameter is best for static and dynamic load balancing. The table 1 describe for compare static and dynamic in performance parameters.

X.CONCLUSION

Load balancing is a process for distributing tasks onto multiple computers. For instance, distributing incoming requests for a web application into multiple web servers. There are a few different ways to implement load balancing. Load balancing algorithms are perfectly dependent upon in which situations nature parameter will compare with static and dynamic load balancing workload is assigned, during compile time and run time. The above comparison shows that static load balancing algorithms are faster than dynamic. But dynamic load balancing algorithms are always better than static as per as adaptability, cooperativeness, fault tolerant, resource utilization, response, waiting time and throughput is concert.

REFERENCE

[1] Palak Shrivastava,Sudheer Kumar Arya , Dr. Priyanka Tripathi, “Various Issues & Challenges of Load Balancing Over Cloud: A Survey”, International Journal Of Engineering And Computer Science, vol 5, ppno: 17517-17524, aug 2016.

[2] Akanksha Mathur, Virender Singh Shekhawat, “Load Balancing in Cloud Computing: A Review”, International Journal of Innovative Research in Computer and Communication Engineering, vol. 3, ppno:11537-11543, nov 2015.

[3] Dharmesh Kashyap, Jaydeep Viradiya, “ A Survey Of Various Load Balancing Algorithms In Cloud Computing”, International Journal of Scientific & Technology Research, vol 3, pp no:115-119, nov 2014.

[4] Komal Purba, Nitin Bhagat, “A Review on Load Balancing Algorithm in Cloud Computing”, SSRG International Journal of Computer Science and Engineering (SSRG-IJCSE), vol1,ppno:19- 23, Dec 2014.

[5] Er. Pooja , Er. Vivek Thapar, " Survey of VM Load Balancing Algorithm in Cloud Environment",International Journal of Computer Science Trends and Technology, vol 4,ppno:123-129, Mar -Apr 2016.

[6] Swati Katoch, Jawahar Thakur, " Load Balancing Algoritms in Cloud Computing Environment: A Review",International Journal on Recent and Innovation Trends in Computing and Communication,vol2, ppno: 2151 – 2156,aug 2014.

[7] Abhijit Aditya, Uddalak Chatterjee and Snehasis Gupta, " A Comparative Study of Different Static and Dynamic Load Balancing Algorithm in Cloud Computing with Special Emphasis on Time Factor", International journal of Current Engineering andTechnology,vol5,ppno:1898-1907,june 2015.

[8] Bhatt Hirenkumar .H, Prof. Hitesh A. Bheda, "An Overview of Load balancing Techniques in Cloud Computing Environments", International Journal Of Engineering And Computer Science, vol 4,ppno 9874-9881, January 2015.

BIOGRAPHY

S. Saranya received her B.SC (Computer Science) from Sree Saraswathi Thyagaraja College, Pollachi, India. She completed her Master of Computer Applications (MCA) from Sree Saraswathi Thyagaraja College, Pollachi, India. Presently, she is a Research Scholar at Department of Computer Science, NGM College, and Pollachi, India. She presented a Research Paper on national Conference. Her area of interest includes Cloud computing, Computer Network, Data Mining.

Dr R. Manicka chezian received his M.Sc. Applied Science from PSG College of Technology, Coimbatore, India in 1987. He completed his M.S. degree in Software Systems from Birla Institute of Technology and Science, Pilani, Rajasthan, India and Ph.D. degree in Computer Science from School of Computer Science and Engineering, Bharathiar University, Coimbatore. He has 25 years of Teaching experience and 17 years of Research Experience. He served as a Faculty of Maths and Computer Applications at P.S.G College of Technology, Coimbatore from 1987 to 1989. Presently, He is working as an Associate Professor of Computer Science in NGM College (Autonomous), Pollachi, India. He has published more than 120 papers in various International Journals and Conferences. He is a recipient of many awards like Desha Mithra Award and Best paper Award. He is a member of various Professional Bodies like Computer Society of India and Indian Science Congress Association. His research focuses on Network Databases, Data Mining, Data Compression, Mobile Computing and Real Time Systems, network,bio-informatics and distributed computing.