IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Volume-4, Issue-04 E-ISSN: 2347-2693

Study of Various Load Balancing Techniques in Cloud Environment- A Review

Rajdeep Kaur[1*] and Amanpreet Kaur[2]

[1,2]Department of Computer Science and Engineering
Global Institute of Management & Emerging Technologies, Amritsar (PUNJAB)

www.ijcseonline.org

Received: Mar/20/2016 Revised: Apr/02/2016 Accepted: Apr/14/2016 Published: Apr/30/ 2016

***Abstract*- Cloud computing is a model for delivering information technology services in which resources are retrieved from the internet through web-based tools and applications, rather than a direct connection to a server. Users can set up and boot the required resources and they have to pay only for the required resources. Thus, in the future providing a mechanism for efficient resource management and assignment will be an important objective of Cloud computing. Cloud Computing is a new trend emerging in IT environment with huge requirements of infrastructure and resources. Load Balancing is an important aspect of cloud computing environment. Efficient load balancing scheme ensures efficient resource utilization by provisioning of resources to cloud user's on-demand basis in pay-as-you-say-manner. Load Balancing may even support prioritizing users by applying appropriate scheduling criteria.**

***Keywords*- Cloud Computing, Virtual Machine, Load balancing, Genetic, Resource scheduling, Service level agreement (SLA)**

I. INTRODUCTION

Cloud Computing is made up by aggregating two terms in the field of technology. First term is Cloud and the second term is computing. Cloud is a pool of heterogeneous resources. It is a mesh of huge infrastructure and has no relevance with its name "Cloud". Infrastructure refers to both the applications delivered to end users as services over the Internet and the hardware and system software in datacenters that is responsible for providing those services. In order to make efficient use of these resources and ensure their availability to the end users "Computing" is done based on certain criteria specified in SLA. Infrastructure in the Cloud is made available to the user's On-Demand basis in pay-as-you-say-manner [1].

Cloud computing refers to the delivery of computing and storage capacity as a service to a heterogeneous community of end-recipients. Cloud computing is an internet technology that utilizes both central remote servers and internet to manage the data and applications. This technology allows many businesses and users to use the data and application without an installation. Users and businesses can access the information and files at any computer system having an internet connection. Cloud computing provides much more effective computing by centralized memory, processing, storage and bandwidth [2].

II. VIRTUAL MACHINE

A Virtual Machine (VM) is a software implementation of a computing environment in which an operating system (OS) or program can be installed and run. The Virtual Machine typically emulates a physical computing environment, but requests for CPU, memory, hard disk, network and other hardware resources are managed by a virtualization layer which translates these requests to the underlying physical hardware. VMs are created within a virtualization layer, such as a hypervisor or a virtualization platform that runs on top of a client or server operating system. This operating system is known as the host OS. The virtualization layer can be used to create many individual, isolated VM environments.

III. VM LOAD BALANCING

Virtual Machine enables the abstraction of an OS and Application running on it from the hardware. The interior hardware infrastructure services interrelated to the Clouds are modeled in the simulator by a Datacenter element for handling service requests. These requests are application elements sandboxed within VMs, which need to be allocated a share of processing power on Datacenter's host components. Data Center object manages the data center management activities such as VM creation and destruction and does the routing of user requests received from user via the Internet to the VMs. The Data Center Controller, uses a VmLoadBalancer to determine which VM should be assigned the next request for processing. Most common VM load balancer are throttled and active monitoring load balancing algorithms. Throttled Load Balancer maintain a record the state of each Virtual Machine (busy/ideal), if a request arrive concerning the allocation of Virtual Machine, throttled load balancer send the ID of ideal Virtual Machine to the data center controller and data center controller allocates the ideal Virtual Machine. Active Monitoring Load Balancer maintains information about each VMs and the number of requests currently allocated to which VM. When a request to allocate a new VM arrives, it identifies the least

loaded VM. If there are more than one, the first identified is selected.

Class diagram of Cloud architecture illustrating relationship between the four basic entities is shown in figure 1. Thus, the object oriented approach of CloudSim can be used to simulate Cloud Computing environment.

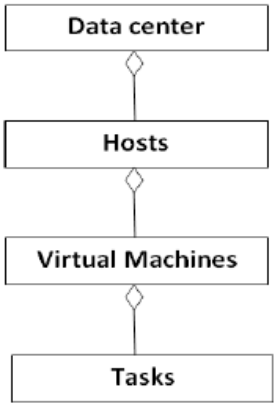

Figure 1- Class diagram of Cloud

Load balancing handles various issues:

- Efficient distribution of user processes on virtual machines.
- Efficient distribution of virtual machines on the physical servers.

Load Balancing helps in:

- Improving the performance substantially.
- Having a Reverse up plan in case the system fails even partially.
- Maintenance of system stability.
- Accommodation of future modification.
- Efficient load distribution.
- Cost effectiveness. [4]

IV. LOAD BALANCING SCHEMES

Load balancing algorithms can be classified into static and dynamic approaches:

A. *Static load balancing algorithm*

Static load balancing algorithms assume that a priori information about all the characteristics of the jobs, the computing resources and the communication network are known and provided. Load balancing decisions are made deterministically or probabilistically at compile time and remain constant during runtime. The static approach is attractive because it is simple and requires minimized runtime overhead. However, it has two major disadvantages. Firstly, the workload distribution of many applications cannot be predicted before program execution. Secondly, it assumes that the computing resources and communication network are all known in advance and remain constant. Such an assumption may not apply to a distributed environment. As static approach cannot respond to the dynamic runtime environment, it may lead to load imbalance on some resources and significantly increase the job response time.

B. *Dynamic load balancing algorithm*

Dynamic load balancing algorithms attempt to use the runtime state information to make more informative decision in sharing the system load. However, dynamic scheme is used a lot in modern load balancing method due to their robustness and flexibility.

A list of common parameters that can be used to characterize most of dynamic load balancing algorithms are:

C. *Adaptive vs. non-adaptive*

If the parameters of the algorithm can change when the algorithm is being run, the algorithm is said to adaptive (to the changes in the environment in which it is running). Otherwise, it is non-adaptive.

a) Sender-initiated vs. receiver-initiated

In a source-initiated algorithm, an over-loaded node starts negotiations with the other nodes for a potential process-migration. If a negotiation is started by an under loaded node, the algorithm is said to be destination-initiated.

b) Preemptive vs. non-preemptive

If a process that has started its execution can be transferred to some other node, then the algorithm is called a preemptive algorithm. If, on the other hand, only those processes that are in the ready queue but have not yet received CPU service could be considered for migration, the algorithm is called a non-preemptive algorith
m.[3]

V. LOAD BALANCING POLICIES

An algorithm for the load balancing problem can be broadly categorized in terms of four policies.They are:

A. *Location policy*

It is the policy that affects the finding of a suitable node for migration. The common technique followed here is polling, on a broadcast, random, nearest-neighbor or roster basis.

B. *Transfer policy*

It is that which determine whether a node is suitable for participating in a process migration. One common technique followed is the threshold policy, where a node participates in a negotiation only when its load is less than (in destination-

initiated algorithm) or greater than (in sender-initiated algorithm) a threshold value.

C. Selection policy

It is the policy that deals with the selection of the process to be migrated. The common factors which must be considered are the cost of migration (communication time, memory, computational requirement of the process, etc.) and the expected gain of migration (overall speedup of the system, etc.).

D. Information policy

It is that component of the algorithm that decides what, how and when the information regarding the state of the other nodes in the system in gathered and managed. They can be grouped under demand-driven, periodic, or state-change-driven policies.

VI. GENETIC ALGORITHM

Genetic Algorithm is search and optimization technique promised on the evolutionary ideas of natural selection and genetics:

A. Selection

Chromosomes are selected from the population to be parents to crossover. The problem is how to select these chromosomes. There are many methods how to select the best chromosomes, for example roulette wheel selection, Boltzman selection, tournament selection, rank selection, steady state selection and some others.

B. Crossover

Crossover is a genetic operator that combines (mates) two chromosomes (parents) to produce a new chromosome (offspring). The idea behind crossover is that the new chromosome may be better than both of the parents if it takes the test characteristics from each of the parents. Crossover occurs during evolution according to a user-definable crossover probability. There are many crossover operator types, for example one point, two point, multi point, arithmetic, heuristic.

C. Mutation

Mutation is a genetic operator that alters one or more gene values in a chromosome from its initial state. This can result in entirely new gene value being added to the gene pool. With these new gene values, the genetic algorithm may be able to arrive at better solution than was previously possible. Mutation is an important part of the genetic search as it helps to prevent the population from stagnating at any local optima. Mutation occurs during evolution according to a user-definable mutation probability. This probability should usually be set fairly low (0.01 is a good first choice). If it is set to high, the search will turn into a primitive random search. There are many mutation operator types, for example, flip bit, boundary, uniform, non-uniform, Gaussian. [3]

VII. LOAD BALANCING ALGORITHMS

The load balancing algorithms are designed to balance the load in the system for the fulfillment of the goals and producing the optimized results as a whole. The important things to consider while developing such algorithm are: estimation and comparison of load, stability of different system, performance of system, interaction between the nodes, nature of work to be transferred, selecting of nodes and many other ones.

The Challenges faced by Load Balancing Algorithms are:

- **Virtual Machines Migration:** How to distribute the data among various machines.
- **Development of small data centres for cloud computing:** Helpful in creation of small and highly beneficial data centres leading to geo-diversity computing.
- **Energy Management:** Helpful in obtaining the economy of the scale.
- **Management of stored data:** Distribution of data for the optimum utilization of space for the storage of data.
- **Automated service provisioning:** Increasing elasticity and provisioning of resources automatically.

Various Load balancing algorithms are:

A. Round Robin

It is a type of Static and Decentralized algorithm in nature. In the following algorithm, the processes are divided between all processors. Each process is assigned to the processor in a round robin order using a particular time value. The process allocation order is maintained locally independent of the allocations from remote processors. Though the work load distributions between processors are equal but the job processing time for different processes are not same. So at any point of time some nodes may be heavily loaded and others remain idle. This algorithm is mostly used in web servers. This algorithm is also implemented as Weighed Round Robin Algorithm. In this we assign the weights to the system in a manner for better resource allocation and utilization.

B. Throttled Load Balancer

This algorithm is a dynamic load balancing algorithm. It is used for load balancing in the case of the virtual machines to be used. Here we first check the index values of all the virtual machine in the system. The request is sent where load balancer parses a table for the allocation of the resources in the system. It assigns the request to a particular load balancer which passes or responds reverse the request to the requester and updates the allocation policy [9]. After the successful allocation of the system the whole process for the de-allocation of the system also starts. This mechanism provides

a greater a higher amount of resource sharing and allocation on a whole in the system resulting in the higher performance and utilization. The throttling threshold maintained generally is 1. It could be modified easily to make the threshold a configurable value.

C. *Active Monitoring Load Balancer*

This load balancing policy attempts to maintain equal workloads on all the available VMs. The algorithm used is quite similar to the throttled case as explained above but with faster and timely checking as well as the accessibility of the resources in the system. The ids for the allocation and de-allocation are specified. The value of the count changes with a new request. It gives the maximum utilization and performance of resources and machines respectively. It is a dynamic allocation algorithm. The important things to consider while developing such algorithm are: estimation and comparison of load, stability of different system, performance of system, interaction between the nodes, nature of work to be transferred, selecting of nodes and many other ones. [4]

VIII. LOAD BALANCING CHALLENGES

a) **Providing services automatically:** A main feature of cloud computing is elasticity; resources can be allocated or released automatically as per the end user's requirement. Here challenge is to how to allocate or release the resources by keeping the system performance same as traditional one.

b) **Virtual Machines Migration:** Virtualization is a mean to create a virtual or replicated form of a device or resource that can be a server, storage device, any network or an operating system to unload a physical machine that is currently heavily loaded. Here main challenge is to dynamically distribute the load from that overload physical machine to some other side by moving virtual machine to avoid bottlenecks in Cloud computing system.

c) **Stored Data Management:** Cloud computing includes large sum of data. Main challenge is to distribute data across the cloud that is stored optimally while maintaining fast access for users.

d) **Usage of small data centers:** Small datacenters can be more beneficial, cheaper and less energy consumer than large datacenter. Load balancing will become a problem then to ensure a maximum response time with an optimal distribution of resources.

e) **Spatial Distribution of the Cloud Nodes:** Some algorithms are designed to be efficient only for closely located nodes where communication delays are comparatively small. Here challenge is to design a load balancing algorithm that can work for spatially distributed nodes. For such nodes there would be some other factors too to be noticed such as the speed of the network links among the nodes, the distance between the user and the processing nodes, and the distances between the nodes involved in providing the cloud services.

f) **Point of Failure:** Load balancing mechanism must be designed in a way to overcome from situations such as **SINGLE POINT FAILURE.** For such situations there are some mechanisms that involve certain pattern that uses a controller for whole system. In such situation if that controller fails, it can cause failure to the whole system. It is also a challenge while making a mechanism for proper load balancing in clouds.

g) **Algorithm Complexity:** Load balancing algorithms are preferred to be less complex and must be easy to understand in terms of implementation and operations. High complexity will cause negative performance issues.

IX. LOAD BALANCING BASED ON SPATIAL DISTRIBUTION OF NODES

Nodes in the cloud are highly distributed. Hence the node that makes the provisioning decision also governs the category of algorithm to be used. There can be three types of algorithms that specify which node is responsible for balancing of load in cloud computing environment.

A. *Centralized Load Balancing*

In centralized load balancing technique all the allocation and scheduling decision are made by a single node. This node is responsible for storing knowledge base of entire cloud network and can apply static or dynamic approach for load balancing. This technique reduces the time required to analyze different cloud resources but creates a great overhead on the centralized node. Also the network is no longer fault tolerant in this scenario as failure intensity of the overloaded centralized node is high and recovery might not be easy in case of node failure.

B. *Distributed Load Balancing*

In distributed load balancing technique, no single nodes responsible for making resource provisioning or task scheduling decision. There is no single domain responsible for monitoring the cloud network instead multiple domains monitor the network to make accurate load balancing decision. Every node in the network maintains local knowledge base to ensure efficient distribution of tasks in static environment and re-distribution in dynamic environment. In distributed scenario, failure intensity of a node is not neglected.

Hence, the system is fault tolerant and balanced as well as no single node is overloaded to make load balancing decision.

C. Hierarchical Load Balancing

Hierarchical load balancing involves different levels of the cloud in load balancing decision. Such load balancing techniques mostly operate in master slave mode. These can be modeled using tree data structure wherein every node in the tree is balanced under the supervision of its parent node. Master or manager can use light weight agent process to get statistics of slave nodes or child nodes. Based upon the information gathered by the parent node provisioning or scheduling decision is made. Three-phase hierarchical scheduling proposed in paper has multiple phases of scheduling. Request monitor acts as a head of the network and is responsible for monitoring service manager which in turn monitor service nodes. First phase uses BTO (Best Task Order) scheduling, second phase uses EOLB (Enhanced Opportunistic Load Balancing) scheduling and third phase uses EMM (Enhanced Min-Min) scheduling.

Algorithm	Knowledge Base	Issues to be addressed	Usage	Drawbacks
Static	Prior knowledge base is required about each node statistics and user requirement	• Response Time. • Resource Utilization. • Scalability. • Power consumption. • Energy utilization. • Throughput.	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 a processor to which load is transferred. • Transfer of task to remote machine. • Information gathering. • Load estimation. • Limiting number of migrations.	Used in heterogeneous Environment.	• Complex. • Time Consuming.
Centralized	Single node or Server is responsible for maintaining the statistics of entire network.	• 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 processes in the network responsible for load balancing store their own local database to make 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 node 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.

Table 1- Comparison of different types of Load Balancing scenarios in Cloud Computing Environment

X. CONCLUSION

Load Balancing is the most important aspect of cloud computing. It helps in reducing the dynamic workload across all the nodes for the achievement of higher user and resource satisfaction. It helps in reducing the overhead, response time and increasing scalability. This paper depicts the concept of cloud computing, virtual machine. This paper elaborates the fundamental concept of load balancing and issues related with it. Various load balancing schemes are represented in this paper. Multiple policies are involved in the conception of load balancing which must be consider while implementing load balancing which are also included along with benefits of load balancing. At the end table signifies the whole concept in which various issues, usage and drawbacks of various algorithms are illustrated.

REFERENCES

[1] Mayanka Katyal, Atul Mishra, "A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment", International Journal of Distributed and Cloud Computing, Volume 1, Issue 2, 2013.

[2] Bhaskar. R, Deepu. S, "Dynamic Allocation Method for Efficient Load Balancing in Virtual Machines for Cloud Computing Environment", Advanced Computing: An International Journal, Vol.3, No. 5, 2012.

[3] Chandrasekaran K., "Load Balancing of Virtual Machine Resources in Cloud Using Genetic Algorithm", Elsevier Publications, pp (156-168), 2013.

[4] S.Sujitha and S. J. Mohana, "Secure Data Storage and Retrieval Using Adaptive Integrity Protocol Model in Cloud Environment", International Journal of Computer Sciences and Engineering, Volume-03, Issue-09, Page No (181-184), Sep -2015, E-ISSN: 2347-2693

[5] Kapil B. Morey, Sachin B. Jadhav, "Grid Computing Approach for Dynamic Load Balancing", International Journal of Computer Sciences and Engineering, Volume-04, Issue-01, Page No (40-42), Jan -2016.

[6] Dharmesh Kashyap, "A Survey of Various Load Balancing Algorithms In Cloud COmputing", International Journal of Scientific & Technology Research, VOLUME 3, ISSUE 11, 2014.

[7] Martina Poulose and M. Azath, "A Study of Load Balancing Techniques in Cloud", International Journal of Computer Sciences and Engineering, Volume-03, Issue-01, Page No (24-27), Jan -2015.

[8] Radha G. Dobale, "Review of Load Balancing for Distributed Systems in Cloud", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 2, 2015.

[9] Gunpriya Makkar, "A Review of Load Balancing in Cloud computing", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 4, 2015.