International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 2 | ISSN : 2456-3307

A Survey on Load Management Techniques in Cloud Computing

M. Priyanka, V. M. Sivagami

Department of Information Technology, Sri Venkateswara College of Engineering, Chennai, Tamil Nadu

ABSTRACT

Cloud computing is emerging as a new standard model for enabling ubiquitous network access, computing resources, deploying, organizing, and accessing vast distributed computing applications over the network. It is an awesome platform in next stage of evolution of internet that leverages various opportunities to improve the way in which we think about and implement the practices and technology needed to secure the things that matters us the most. In cloud computing, Load balancing is one of the main challenges which are required to distribute the workload equally across all the nodes. Load balancing uses services offered by many computer network service provider corporations. The load can be CPU load, memory, capacity, delay or network load. Load balancing ensures that all the processor in the system or every node in the network distributes equal amount of work at any instant of time. This paper is a brief discussion on different load balancing techniques on the bases of different load balancing metrics.

Keywords: Cloud Computing; Load Balancing, Load Balancing Metrics

I. INTRODUCTION

Cloud is a parallel and large scale distributed computing system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service level agreements (SLA) established through negotiation between the service provider and consumers. It has moved computing and data away from desktop and portable PCs, into large data centers. It has the capability to harness the power of Internet and wide area network (WAN) to use resources that are available remotely, thereby providing cost- effective solution to most of the real life requirements. It provides the scalable IT resources such as applications and services, as well as the infrastructure on which they operate, over the Internet, on pay-per-use basis to adjust the capacity quickly and easily. It helps to accommodate changes in demand and helps any organization in avoiding the capital costs of software and hardware. Thus, cloud computing is a framework for enabling a suitable, on-demand network access to a shared pool of computing resources (e.g. networks, servers, storage, applications, and services). These resources can be provisioned and deprovisioned quickly with minimal management effort or service provider interaction. Due to the exponential growth of cloud computing, it has been widely adopted by the industry and there is a rapid expansion in data-centers. This expansion has caused the dramatic increase in energy use and its impact on the environment in terms of carbon footprints. The link between energy consumption and carbon emission has given rise to an energy management issue which is to improve energy-efficiency in cloud computing to achieve Green computing. Besides this, there are many other existing issues like Load Balancing, Virtual Machine Migration, Server Consolidation etc. that are not yet fully addressed. Virtual Machine Migration enabled by virtualization can help in balancing load, enabling highly responsive provisioning and avoiding hot-spots in data centers thereby reducing power consumption. Server Consolidation helps in improving resource utilization by consolidating various VMs residing on multiple under-utilized servers onto a single server, thereby turning off unused servers, hence reducing energy consumption in a cloud computing environment.

The rest of the paper is organized as follows; Section II is about features and load balancing methods in cloud computing, In Section III related work were discussed and the paper is concluded in Section IV.

II. FEATURES AND METHODS

1. Features of Cloud Computing

- ✓ Self-service provisioning means the computing resources are used for almost any kind of work.
- ✓ Flexibility refers to changes in usage to increase or decrease as users see their needs change.
- ✓ Pay-per-use allows users to pay for only the services and resources they use.
- ✓ Maintenance of the servers is easy. Servers are off site, and the providers take care of software updates. There is no need for users to install software on to individual systems and keep up to date with the software as it develops.
- ✓ Location-free allows users to work from anywhere. As long as you have an Internet connection you can access the cloud. Many providers often have apps available to multiple devices, so you are not restricted. More and more businesses are now granting employees the ability to work away from the office. This feature allows a work-life balance and maintains productivity rates.
- ✓ Backup: Cloud-based backup saves time, avoids larger investments and the need to recruit third-party experts. Choosing cloud backups decrease the size of their data centers. The lowering of the numbers of servers, the software expense, and the staff personnel can significantly reduce IT costs without negatively affecting an organization's IT capabilities.
- ✓ Cloud-based workload and file sharing streamline teamwork collaboration, which allows data to be shared, accessed and edited from anywhere. Updates are visible in real-time and are seen by all teammates.
- ✓ Reliability: Reliability is improved by having multiple sizes for the same service, such that if one faces an outage, the other can take over load for the time being.
- ✓ Consumption based billing: Pay per use seems to be the winning characteristic for cloud.
- ✓ Safety: With every one of these elements, safety must be a concern and password protection.

2. Load Balancing

In cloud environment, Load balancing is a technique that distributes the excess dynamic local workload evenly across all the nodes. Load balancing focuses on maximum throughput, avoid overloading and reducing energy consumption by evenly distributing the load, minimizes response time, reduces network latency. Load balancing takes account of minimum resource consumption by two things, one is the resource provisioning or resource allocation and other is task scheduling in distributed environment. The importance of load balancing is estimation of load, load comparison, system stability, system performance, interaction between the nodes, nature of work to be transferred, selection of nodes and many more to be considered while developing.

Load balancing is a general term for various distribution techniques that help spread traffic and workload across different servers within a network. Put in human terms, the idea is simple: the more available hands working, the faster and more efficiently the job gets done, and the less work each person has to do. When applied to the computer networks, these principles of "community labour" become extremely valuable, as they help increase overall computing efficiency - minimizing downtime and raising overall throughput and performance. As more and more computing is done online, load balancing has taken on a broader meaning. Global Server Load Balancing is the same in principle but its implementation is not confined to one local network. The workload is still distributed, but it's distributed planet-wide instead of just across a data center. As a result, modern-day solutions face new challenges, as they are required to take into account not just an individual cluster of servers, but also communications parameters (e.g. link quality) and geographical location of remote requesters. Today, as more and more online businesses seek to leverage Content Delivery Networks (CDNs), load balancing has become a key component in most content distribution tasks.

Load balancing algorithm is classified into two categories such as static and dynamic load balancing.

- ✓ Static load balancing algorithm: In static load balancing algorithm load balancer uses priori knowledge of the applications and statistical information about the system and distributes the load equivalently between servers.

- ✓ Dynamic load balancing algorithm: Dynamic load balancing algorithms are those algorithms which search for the lightest server in the network and then designates appropriate load on it. In this, work load is distributed among the processors at runtime.

2.1. Metrics for Load Balancing In Cloud

- Overhead Associated - determines the amount of overhead involved while implementing a load-balancing algorithm. It is composed of overhead due to movement of tasks, inter-processor and inter-process communication. This should be minimized so that a load balancing technique can work efficiently.
- Throughput - is used to calculate the no. of tasks whose execution has been completed. It should be high to improve the performance of the system.
- Performance – is used to check the efficiency of the system. It has to be improved at a reasonable cost e.g. reduce response time while keeping acceptable delays.
- Resource Utilization - is used to check the utilization of resources. It should be optimized for an efficient load balancing.
- Scalability - is the ability of an algorithm to perform load balancing for a system with any finite number of nodes. This metric should be improved.
- Response Time - is the amount of time taken to respond by a particular load balancing algorithm in a distributed system. This parameter should be minimized.
- Fault Tolerance - is the ability of an algorithm to perform uniform load balancing in spite of arbitrary node or link failure. The load balancing should be a good fault-tolerant technique.
- Migration time - is the time to migrate the jobs or resources from one node to other. It should be minimized in order to enhance the performance of the system.

2.2. Load Balancing Working Principle

Load balancing enables network administrators to spread work across multiple machines, exploiting available resources more efficiently.To implement such solutions, administrators generally define one IP address and/or DNS name for a given application, task, or web site, to which all requests will come. This IP address or DNS name is actually, of course, the load balancing server.The administrator will then enter into the load balancing server the IP addresses of all the actual servers that will be sharing the workload for a given application or task. This pool of available servers is only accessible internally, via the load balancer. Finally, your load balancer needs to be deployed - either as a proxy, which sits between your app servers and your users worldwide and accepts all traffic, or as a gateway, which assigns a user to a server once and leaves the interaction alone thereafter. Once the load balancing system is in place, all requests to the application come to the load balancer, and are redirected according to the administrator’s preferred algorithm.

2.3. Load Balancing Algorithms

A load balancing algorithm controls the distribution of incoming requests to your cluster of servers. There are numerous methods employed to accomplish this, depending on the complexity of load balancing required, the type of task at hand, and the actual distribution of the requests coming in. Some common methods include:

Round Robin - the most basic load distribution technique, and considered rather primitive by network administrators.
In a round robin scenario the load balancer simply runs down the list of servers, sending one connection to each in turn, and starting at the top of the list when it reaches the end.

Weighted Round Robin – the same principle as Round Robin, but the number of connections that each machine receives over time is proportionate to a ratio weight predefined for each machine.
For example, the administrator can define that Server X can handle twice the traffic of Servers Y and Z, and thus the load balancer should send two requests to Server X for each one request sent to Servers Y and Z. However, given that most enterprises use servers that are uniform in their processing power, Weighted Round Robin essentially attempts to address a nonexistent problem.

Least Connections – transfers the latest session to the server with the least connections at the time of session initiation. To avoid latency, this method is advisable in an environment where server capacity and resources are uniform. Least Connections is considered problematic, as most implementations are challenged to accurately measure actual server workload.

Weighted Least Connections – identical to Least Connection, except that servers are selected based on capacity, not just availability. For each node, the admin specifies a Connection Limit value, and the system creates a proportional algorithm on which load balancing is based. Similar to Weighted Round Robin, this method presumes that server resources are not uniform – which is not in-line with most enterprise network topography.

Least Pending Requests – the emerging industry standard, Least Pending Requests selects the server with the least active sessions based on real-time monitoring. Requires assignment of both Layer 7 and TCP profile to the virtual server. The Least Pending Requests (LPR) algorithm, which is enabled by Layer 7 solution, is currently considered a best practice, as it provides best indication of actual load delivered from each connection.

2.4. DNS Load Balancing

In a DNS scenario, load balancing pools for various geographic regions are established, so the load balancer knows exactly which web servers are available for traffic and how often they should receive traffic. This enables the administrator to take advantage of geographically dispersed infrastructure and enhance performance by shortening the distance between requesters and data centers. Although, in some specific scenarios, DNS load balancing can be effective for simpler applications or web sites, it has notable limitations, which lower overall efficacy for mission-critical deployments.DNS load balancing uses a simple Round Robin methodology. Unfortunately, DNS records have no native failure detection. This means that if the next server in the rotation is down, requesters will be directed to it anyway – unless the organization adopts a third-party monitoring solution, which adds yet another source of implementation, configuration and maintenance complexity. Moreover, a DNS solution cannot take into account the unknown percentage of users who have DNS data cached, with varying amounts of Time to Live (TTL) left. And so, until TTL times out, visitors may still be redirected to the "wrong" server.

2.5. High Overhead of Hardware Load Balancers

Until recently, most hardware load balancing was based on a hardware load-balancing device (HLD). Also known as a layer 4-7 router, an HLD is an actual physical unit in the network, which works by directing users to individual servers based on various usage parameters such as server processor utilization, number of connections to a server, and overall server performance. Today, single-function HLDs are being replaced by multi-function ADCs (application delivery controllers). An ADC delivers a full range of functions that optimize enterprise application environments, including load balancing, reliability, data center resource use, end-user performance, security, and more. ADC-based hardware is server-based (as opposed to content-switch-based). Server-based load balancing leverages standard PC-class servers on which special load-balancing software has been installed. Content-switch-based load balancers are actual network switches that have load-balancing software on-board, and act as intelligent switching devices. Moreover, of the primary problems with load-balancing ADCs is that they can represent a single point of failure, and can bottleneck traffic if not configured or maintained properly. Also, the setup for ADCs is complex, requiring dedicated and expert staff. Hardware solutions can be effective for organizations with the resources to manage the complex installation process, the high maintenance overhead, and ongoing capital outlays associated with hardware. However, HLDs and ADCs alike are aging technology, considered unnecessarily costly and resource-intensive by forward-thinking network admins.

2.6. Software Load Balancing

Software load balancing, as the name implies, is based on software solutions and as such is mostly independent of the platform on which the load balancing utility is installed. Such software can be implemented either as an add-on application bundled with a DNS solution, as part of an operating system, or – more commonly today - as part of virtual service and application delivery solution. Service and application delivery solutions are designed to optimize, secure and control the delivery of all enterprise and cloud services, maximizing end user experience for all users, including mobile users. Most of these packages include an integral virtualized load balancing solution. However, no matter how it is implemented, software-based solutions ultimately requires either hardware to run on or intensive setup and maintenance. This forces organizations to work with multiple vendors, often leading to compatibility issues which are impractical for larger organizations. Thus, although software-based load balancing solutions may appear less expensive than hardware-based solutions, Total Cost of Ownership (TCO) for software solutions is still high - and setup and maintenance tasks are no less demanding. For this reason, software-based solutions – virtual or not - are often beyond the means of many SMEs and are generally not the first choice for other organizations, as well.

III. RELATED WORK

Fahimeh Farahnakian et al. [3] addresses VM consolidation problem with the objective to reduce energy consumption of data centers while satisfying QoS requirements. A distributed system architecture for dynamic VM consolidation to improve resource utilizations of PMs and to reduce their energy consumption is presented. It also proposed a dynamic VM consolidation approach that uses a highly adaptive online optimization meta heuristic algorithm called Ant Colony System (ACS) to optimize VM placement. The proposed ACS-based VM Consolidation (ACS-VMC) approach uses artificial ants to consolidate VMs into a reduced number of active PMs according to the current resource requirements. These ants work in parallel to build VM migration plans based on a specified objective function

J. Octavio et al. [4] contributes distributed problem solving techniques for load management in data centers supported by VM live migration. Collaborative agents are end owed with a load balancing protocol and an energy-aware consolidation protocol (EProtocol) to balance and consolidate heterogeneous loads (e.g., migrating memory-intensive loads to memory intensive hosts) in a distributed manner while reducing energy consumption costs. When a host is overloaded, agents collaborate with each other to sample resource usage of hosts and determine the best destination host for a VM according to server-centric load management policies. In addition, hosts are managed by server manager agents (SMAs), which collaborate among each other to consolidate VMs (deployed in potentially underutilized hosts) into fewer hosts. A server manager agent deployed in an underutilized host interacts with a front-end agent in order to be designated as a leader, meaning the server manager agent can start migrating VMs to other hosts and autonomously turning itself off afterward. Additionally, agents deployed in front-end servers autonomously turn on hosts when VM allocation requests cannot be fulfilled due to insufficient resources.

Medhat Tawfeek et al. [5] proposed ACO algorithm to find the optimal resource allocation for tasks in the dynamic cloud system to minimize the make span of tasks on the entire system. Then, this scheduling strategy was simulated using the cloud sim toolkit package. Experimental results compared to First Come First Serve (FCFS) and Round Robin (RR) shows that ACO algorithm satisfies expectation.

M. Rahman et al. [6] have given the concept of Load Balancer as a Service (LBaaS) in Cloud Computing as their related work shows that researchers have yet not focused on this problem domain. Therefore initially they have focused on load balancing problem, importance of load balancing, expected characteristics in cloud computing and finally on the Load Balancer as a Service in cloud computing. This service model (LBaaS) was highly adopted by market player.

Randles, M et al. [7] called Honeybee Foraging for load balancing in distributed scenario. In Honeybee foraging the movement of ant in search of food forms the basis of distributed load balancing in cloud computing environment. This is a self organizing algorithm and uses queue data structure for its implementation.

Santanu Dam et al. [8] has proposed a method which is a combination of two methods one is GA and other is Gravitational Emulation Local Search (GELS). GELS is used for initiating Population for GA. Initial population is generated based on velocity calculation of chromosome done by GELS. Then, the two chromosomes are selected based on Fitness and two point crossover and Mutation is applied. Again velocity is calculated for chromosome and fitness chromosomes are added to new population. By using Cloud analyst simulation of the proposed algorithm is carried out and satisfactory results are obtained by authors. No priority is considered for the request by the authors which cannot be the real scenario.

S. Sharma et al. [9] presented various approaches of load balancing on the basis of different parameters. This research work also gives the direction to design a new algorithm on the basis of different parameter by analyzing the behaviour of various existing algorithms. At the end of the research work author concluded that it is easy to understand the behaviour of static algorithms as compared to dynamic load balancing algorithm.

S.G.Domanal et al.[10] proposed a hybrid scheduling algorithm for load balancing in a distributed environment by combining the methodology of Divide-and-Conquer and Throttled algorithms referred to as DCBT. This algorithm plays an important role in distributing the incoming load in an efficient manner so that it maximizes resource utilization in a cloud environment. Further, load balancer plays an important role in cloud environment by assigning incoming tasks to Virtual Machines (VM) intelligently. The main aim of the proposed DCBT is to reduce the total execution time of the tasks and thereby maximizing the resource utilization. Further, the proposed DCBT algorithm is analyzed using Cloud Sim simulator and also in customized distributed environment using python. Experimental results demonstrate that the proposed algorithm gives better efficiency in both Cloud Sim and customized environments. The proposed DCBT utilizes the Virtual Machines more efficiently while reducing the execution time of the tasks allocated to Request Handlers (RH) by 9.972% in comparison to the Modified Throttled algorithm.

Nayandeep Sran et al.[11] proposed a virtual graph which is constructed with the connectivity of each node representing the load on server. Each node is represented as a vertex in a directed graph and each in-degree represents free resources of that node. Whenever a client sends a request to the load balancer, the load balancer allocates the job to the node which has atleast one indegree. Once a job is allocated to the node, the indegree of that node is decremented by one. After the job is completed, the node creates an incoming edge and increments the in-degree by one. The addition and deletion of processes is done by the process of random sampling. Each process is characterized by a parameter know as threshold value, which indicates the maximum walk length. A walk is defined as the traversal from one node to another until the destination is found. At each step on the walk, the neighbour node of current node is selected as the next node. In this algorithm, upon receiving the request by the load balancer, it would select a node randomly and compares the current walk length with the threshold value. If the current walk length is equal to or greater than the threshold value, the job is executed at that node. Else, the walk length of the job is incremented and another neighbour node is selected randomly. The performance is degraded as the number of servers increase due to additional overhead for computing the walk length.

Benifa et al. [13] proposed a scheduling strategy named efficient locality and replica-aware scheduling (ELRAS) integrated with an autonomous replication scheme (ARS) to enhance the data locality and performs consistently in the heterogeneous environment. ARS autonomously decides the data object replicated by considering its popularity and removes the replica as it is idle. The results proved that efficiency of the algorithm is better for heterogeneous clusters and workloads.

Singh et al. [14] proposed a novel autonomous agent-based load-balancing algorithm called A2LB for cloud environments. Their algorithm tries to balance the load among VMs through three agents: load agent, channel agent, and migration agent. Load and channel agents are static agents whereas migration agent is an ant, which is a special category of mobile agents. Load agent controls the information policy and calculates a load of VMs after allocating a job. A VM Load Fitness table supports the load agent. The fitness table maintains the list of all details of the VM properties in a data center such as id, memory, a fitness value, and load status of all VMs. Channel agent controls the transfer policy, selection policy, and location policy. Finally, the channel agent initiates the migration agents. They move to other data centers and communicate with the load agent of that data center to acquire the status of VMs present there, looking for the desired configuration. Result obtained through implementation proved that this algorithm works satisfactorily.

IV. CONCLUSION AND FUTUREWORK

Thus in cloud environment, proper load balancing avoids fail-over and bottlenecks which in turn improves flexibility, scalability of resources and it also reduces over-provisioning of VM allocation and minimizes resource utilization. This paper presents a detailed survey on load balancing algorithms along with the metrics.

In future, research work will be carried out in conservation of energy on cloud and also workload optimization by providing efficient computation and effective processing of resources.

V. REFERENCES

[1]. Bei Guan, Jingzheng Wu, Yongji Wang, and Samee U. Khan, Senior Member, IEEE "CIVSched: A Communication-aware Inter-VM Scheduling Technique for Decreased Network Latency between Co-located VMs" in IEEE transactions on cloud computing 2168-7161 (c) 2013 IEEE

[2]. Ali M. Alakeel, "A Guide to Dynamic Load Balancing in Distributed Computer Systems", IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.6, June 2010.

[3]. Fahimeh Farahnakian, Member, IEEE, Adnan Ashraf, Tapio Pahikkala, Member, IEEE, Pasi Liljeberg, Juha Plosila, Ivan Porres, and Hannu Tenhunen, Member, IEEE,"Using Ant Colony System to Consolidate VMs for Green Cloud Computing" in IEEE transactions on services computing, vol. 8, no. 2, March/April 2015.

[4]. J.Octavio Gutierrez-Garcia and Adrian Ramirez-Nafarrate,"Collaborative Agents for Distributed Load Management in Cloud Data Centers Using Live Migration of Virtual Machines," in IEEE transactions on services computing, vol. 8, no. 6, November/December 2015.

[5]. Medhat Tawfeek, Ashraf El-Sisi, Arabi Keshk and Fawzy Torkey Faculty of Computers and Information, Menoufia University, Egypt "Cloud Task Scheduling Based on Ant Colony Optimization " in The International Arab Journal of Information Technology, Vol. 12, No. 2, March 2015.

[6]. Rahman, M., Iqbal, S., & Gao, J., "Load Balancer as a Service in Cloud Computing", IEEE 8th International Symposium on Service Oriented System Engineering, pp. 204-211, April 2014.

[7]. Randles, M., Bendiab, A. T. & Lamb, D. (2008). Cross layer dynamics in self-organising service oriented architectures. IWSOS, Lecture Notes in Computer Science, 5343, pp. 293-298, Springer.

[8]. Santanu Dam, Gopa Mandal, Kousik Dasgupta and Paramartha Dutta,Genetic Algorithm and Gravitational Emulation Based Hybrid Load Balancing Strategy in Cloud Computer, Communication, Control and Information Technology (C3IT), 2015 Third International Conference 2015, IEEE 2015.

[9]. Sharma S., Singh S., & Sharma, M. "Performance Analysis Of Load Balancing Algorithms" World Academy of Science, Engineering and Technology, 38, pp. 269-272, 2008.

[10]. S. G. Domanal and G. R. M. Reddy, "Load Balancing in Cloud Environment Using a Novel Hybrid Scheduling Algorithm," 2015 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), Bangalore, 2015, pp. 37-42.

[11]. Nayandeep Sran, Navdeep kaur , Comparative Analysis of Existing Load balancing techniques in cloud computing, International Journal of Engineering Science Invention, Vol-2 Issue-1 2013.

[12]. Stuti Dave, Prashant Mehta "Utilizing Round Robin Concept for Load Balancing Algorithm at Virtual Machine Level in Cloud Computing" IJAC (0975-8887) Volume 94-No.4, May 2014.

[13]. Benifa, JVB. and Dejey (2017). Performance Improvement of MapReduce for Heterogeneous Clusters Based on Efficient Locality and Replica Aware Scheduling (ELRAS) Strategy. Wireless Personal Communications, 1-25.

[14]. Singha, A. and Juneja, D., and Malhotra, M. (2015). Autonomous Agent Based Load-balancing algorithm in Cloud Computing. International Conference on Advanced Computing Technologies and Applications (ICACTA), 45, 832–841