Volume 4, Issue 3, March 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Survey on Load Balancing in Public Cloud

V.V.Divya Bharathi, S.Nivedha, T.Priyanka
IT & Anna University
India

Asst.Prof Mrs. M.Daya Kanimozhi Rani, M.E.,(Ph,D).,
Department of Information Technology
Adhiyamaan College of Engineering, India

ABSTRACT: Cloud computing is a rapid growing domain and more users are attracted towards utility computing, better and fast service needs to be provided. For better management of available resources, good load balancing techniques are required. In this paper, load balancing model divides the public cloud into several cloud partitions using switch mechanism. The improved round robin algorithm and game theory concept is used to maintain load and provide better strategies through efficient job scheduling and resource allocation techniques as well.

Keywords: Cloud computing, Cloud partitioning, Improved round robin algorithm, Game theory.

I. INTRODUCTION

Load Balancing

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction[1]. Through cloud computing there is no need to store the data on desktops, portables etc. You can store the data on servers and you can access the data through internet. The main issue related to cloud computing is load balancing. Load occurs when the number of job increases. Load Balancing is a technique in which the workload on the resources of a node is shifts to respective resources on the other node in a network without disturbing the running task. The load can be a memory, CPU capacity, network or delay load. Load balancing is the process of reassigning the total loads to the individual nodes of the collective system to make the best response time and also good utilization of the resources. In this paper we use cloud partitioning with switch mechanism to balance the load in the public cloud. The cloud can be partition either by using static or dynamic parameters. The static parameters are CPU processing speed, queue size, memory size. The dynamic parameters are memory utilization ratio, CPU utilization

**Department Of Information Technology,*
Adhiyamaan College Of Engineering-Hosur

ratio, network bandwidth, etc. The main goals of load balancing are to optimize time and avoid load.

II. TYPES OF LOAD BALANCING

Generally, load balancing algorithm can be classified into two major categories depending on how the process is allocated to the nodes (system load) and status information of the node (system topology).

2.1 Classification According to the System Load

Centralized approach: In this approach, a single node is responsible for managing the distribution within the whole system.

Distributed approach: In this approach, each node independently builds its own load vector by collecting the load information of other nodes. Decisions are made locally using local load vectors. This approach is more suitable for widely distributed systems such as cloud computing.

Mixed approach: A combination between the two approaches to take advantage of each approach.

2.2 Classification According to the System Topology

Static approach: This approach is generally defined in the design or implementation of the system.

Dynamic approach: 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.

Adaptive approach: This approach adapts the load distribution to system status changes, by changing their parameters dynamically and even their algorithms. This approach is able to offer better performance when the system state changes frequently. This approach is more suitable for widely distributed systems such as cloud computing.

2.3 Existing Load Balancing Technique

The existing load balancing techniques in cloud computing are

Event-driven- V. Nae et al.[1] presented an event-driven load balancing algorithm for real time massively multiplayer online games (MMOG). This algorithm after receiving capacity events as input, analyzes its components in context of the resources and the global state of the game session, thereby generating the game session load balancing actions. It is

capable of scaling up and down a game session on multiple resources according to the variable user load but has occasional QoS breaches.

Lock-free multiprocessing solution for LB- X. Liu et al. [2] proposed a lock-free multiprocessing load balancing solution that avoids the use of shared memory in contrast to other multiprocessing load balancing solutions which use shared memory and lock to maintain a user session. It is achieved by modifying Linux kernel. This solution helps in improving the overall performance of load balancer in a multi-core environment by running multiple load-balancing processes in one load balancer

Honeybee Foraging Behaviour- M. Randles et al. [4] investigated a decentralized honeybee-based load balancing technique that is a nature-inspired algorithm for self-organization. It achieves global load balancing through local server actions. Performance of the system is enhanced with increased sys- tem diversity but throughput is not increased with an increase in system size. It is best suited for the conditions where the diverse population of service types is required.

Join-Idle-Queue- Y. Lua et al. [1] proposed a Join-Idle- Queue load balancing algorithm for dynamically scalable web services. This algorithm provides large-scale load balancing with distributed dispatchers by, first load balancing idle processors across dispatchers for the availability of idle processors at each dispatcher and then, assigning jobs to processors to reduce average queue length at each processor. By removing the load balancing work from the critical path of request processing, it effectively reduces the system load, incurs no communication overhead at job arrivals and does not increase actual response time

Active Clustering- M. Randles et al. [4] investigated a self- aggregation load balancing technique that is a self-aggregation algorithm to optimize job assignments by connecting similar services using local re-wiring. The performance of the system is enhanced with high resources thereby in- creasing the throughput by using these resources effectively. It is degraded with an increase in system diversity.

Open Flow model- HardeepUppal presented a model in which open flow Techniques of load balancing in cloud computing: A survey switch is used. Open flow switches are like a standard switch with a flow table performing packet lookup and forwarding. The difference lies in how flow rules are inserted and updated inside the switch's flow table [3].

Ant Colony Optimization- Z. Zhang et al. [6] proposed a load balancing mechanism based on ant colony optimization in an open cloud computing federation. It uses small- world and scale-free characteristics of a complex network to achieve better load balancing. This technique overcomes heterogeneity, is adaptive to dynamic environments, is excel-lent in fault tolerance and has good scalability hence helps in improving the performance of the system.

VectorDot- A. Singh et al. [5] proposed a novel load balancing algorithm called VectorDot. It handles the hierarchical complexity of the data centre and multidimensionality of resource loads across servers, network switches, and storage in an agile data centre that has integrated server and storage virtualization technologies. VectorDot uses dot product to distinguish nodes based on the item requirements and helps in removing overloads on servers, switches and storage nodes.

Biased random sampling- M. Randles et al.[4] 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. The performance of the system is improved with high and similar population of resources thus resulting in an increased throughput by effectively utilizing the increased system resources.

2.4 Existing Load Balancing Algorithms

Throttled Load Balancing Algorithm

Throttled algorithm is completely based on virtual machine. In this client first requesting the load balancer to check the right virtual machine which access that load easily and perform the operations which is given by the client or user. This ensures that only a pre-defined number of internet cloud-lets are allocated to a single VM at any given time. If more request groups are present than the number of available VMs at a data centre, some of the request will be queued until the next VM becomes available Throttled algorithm is completely based on virtual machine. In this client first requesting the load balancer to check the right virtual machine which access that load easily and perform the operations which is given by the client or user

Equally Spread Current Execution

Equally load distributing improves performance by transferring load from heavily loaded server. Efficient scheduling and resource allocation is a critical characteristic of cloud computing based on which the performance of the system is estimated. In spread spectrum technique load balancer makes effort to preserve equal load to al the VMs connected with the data centre. Load balancer maintains an index table of VMs as well as number of requests currently assign to the VM. If the request comes from the data centre to allocate a new VM, it scans the index table for the least loaded VM

Round Robin

It is the simplest algorithm that uses the concept of time quantum or slices. Here the time is divided into multiple slices and each node is given a particular time quantum or time interval and in this quantum the node will perform its operations. The resources of the service provider are provided to the client on the basis of this time quantum.

III. IMPROVED ROUD ROBIN ALGORITHM AND GAME THEORY

Improved Round Robin Algorithm

Round Robin scheduling algorithm is the widely used scheduling algorithm in multitasking and real time environment. It is the most popular algorithm due to its fairness and starvation free nature towards the processes, which is achieved by

using the time quantum. As the time quantum is static, it causes less context switching in case of high time quantum and high context switching in case of less time quantum. Increasing context switch leads to high avg. waiting time, high average turnaround time which is an overhead and degrades the system performance. So, the performance of the system solely depends upon the choice of optimal time quantum which is dynamic in nature. In this paper, we have proposed a new variant of RR scheduling algorithm known as Improved Round Robin (IRR) Scheduling algorithm, by arranging the processes according to their shortest burst time and assigning each of them with an optimal time quantum which is able to reduce all the above said disadvantages. Experimentally we have shown that our proposed algorithm performs better than the RR algorithm, by reducing context switching, average waiting and average turnaround time.

Game Theory

Game theory has non-cooperative games and cooperative games. In cooperative games, the decision makers eventually come to an agreement which is called a binding agreement. Each decision maker decides by comparing notes with each other's. In non-cooperative games, each decision maker makes decisions only for his own benefit.

IV. CONCLUSION

Considering the growing importance of cloud, finding new ways to improve cloud services is an area of concern and research focus. In this paper, we have surveyed various load balancing techniques for cloud computing. The main purpose of load balancing is to satisfy the customer requirement by distributing load dynamically among the nodes and to make maximum resource utilization by reassigning the total load to individual node. When node is overloaded with job at that time load balancer has to set that load on another free node. So, to balance the load is necessary in cloud computing. So in our paper we have discussed all the existing techniques for Load balancing. And we have also discussed the virtualization and cloud computing.

FUTURE WORK

Our future work will encompass testing the proposed algorithm on a Eucalyptus based private cloud, testing the algorithm on more powerful hardware, exploring more efficient VM load balancing and service load balancing (service brokerage) algorithms, and incorporating failure handling mechanisms into the simulation. Our proposed algorithm can be extended using soft and hard real time systems. So in near future we can improve time sharing system by using this algorithm.

REFERENCE

[1]. Nae V., Prodan R. and Fahringer T. (2010) 11th IEEE/ACM International Conference on Grid Computing (Grid), 9-17.

[2]. Liu Xi., Pan Lei., Wang Chong-Jun. and Xie Jun-Yuan. (2011) 3rd International Workshop on Intelligent Systems and Appli- cations, 1-4.

[3]. Zenon Chaczko, Venkatesh Mahadevan, Shahrzad Aslanzadeh and Christopher Mcdermid, Availabilty and load balancing in cloud computing, 2011 International Conference on Computer and Software Modeling, IPCSIT vol.14 (2011) ACSIT Press, Singapore.

[4]. Randles M., Lamb D. and Taleb-Bendiab A. (2010) 24th Inter- national Conference on Advanced Information Networking and Applications Workshops, 551-556.

[5]. Singh A., Korupolu M. and Mohapatra D. (2008) ACM/IEEE conference on Supercomputing.

[6]. Ratan Mishra, Anant jaiswal, Ant colony optimization: A Solution of load balancing in cloud, International J1ournal of Web & Semantic Technology (IJWesT) Vol.3, No.2, April 2012

[7]. Gaochao Xu, Junjie Pang, and Xiaodong Fu "A Load Balancing Model Based on Cloud Partitioning for the Public Cloud" Tsinghua Science And Technology ISSN 1007 - 0214 04 /12 Volume 18, Number 1, February 2013, pp 34-39

[8]. S. Aote and M.U.Kharat, A game-theoretic model for dynamic load balancing in distributed systems, in Proc. the International Conference on Advances in Computing, Communication and Control (ICAC3 '09), New York, USA, 2009, pp. 235-238.

[9]. D. Grosu, A.T.Chronopoulos, and M.Y.Leung, Load balancing in distributed systems: An approach using cooperative games, in Proc. 16th IEEE Intl. Parallel and Distributed Processing Symp., Florida, USA, Apr. 2002, pp. 52-61.

