International Journal of Combined Research & Development (IJCRD)
eISSN: 2321-225X;pISSN:2321-2241 Volume: 5; Issue: 2; February -2016

Load Balancing Model for Performance Enhancement in Public Cloud using Cloud Partitioning

Anisha Kunjan S
Assistant Professor
CMR Institute of Technology
Karnataka, India

Sunitha Sooda
Assistant Professor
HKBK College of Engineering
Karnataka, India

Archana Homalimath
Assistant Professor
HKBK College of Engineering
Karnataka, India

Abstract *: In recent days, cloud computing has improvised the areas of research an industry, which includes distributed computing, internet, web services and virtualization. Load balancing is one of the biggest challenges in cloud computing which is required to distribute the dynamic workload evenly among multiple nodes and also to ensure that no single node is over taken by the workload. Load balancing is an important aspect of cloud computing and has an important impact on the performance.This paper intends to give a better load balancing stategy for the public cloud using the cloud partitioning concept.This cloud partitioning would be provided with a change mechanism for choosing different strategies for different situtations.The algorithm applies the random allocation for load balancing strategy to which ultimately helps to improve the different performance parameters like throughput, response time in the public cloud.*

Keywords: Load Balancing model, cloud partioning, Random allocation.

1. INTRODUCTION

Cloud computing services can be used from diverse and widespread resources, rather than remote servers or local machines. Generally it consists of a bunch of distributed servers known as masters, providing demanded services and resources to different clients known as clients in a network with scalability and reliability of datacenter. The distributed computers provide on-demand services[4].We know that a Cloud system consists of three major components such as clients, datacenter, and distributed servers. Each element has a definite purpose and plays a specificrole[15][17].

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 overloaded while some others are ladened [20][24].A load balancing algorithm used for balancing purpose which is dynamic in nature does not consider the previous state or behavior of the system, that is, it depends on the present behavior of thesystem[16].

Also load balancing is a relatively new technique that facilitates networks and resources by providing a maximum throughput with minimum response time. Proper load balancing can help in utilizing the available resources optimally. Load Balancing is done with the help of load balancers where each incoming request is redirected and is transparent to client who makes the request. Also load balancers may have a variety of specialfeatures[19].

This model divides the public cloud into several cloud partitions. When the environment is very large and complex, these divisions simplify the load balancing. The cloud has a main controller that chooses the suitable partitions for arriving jobs while the balancer for each cloud partition chooses the best load balancing strategy main controller and balancer helps to balance the load and to improve the efficiency.

2. LITERATURE SURVEY

Load balancing in cloud computing was described by B.Adler[7][17] in his white paper "Load balancing in the cloud: Tools, tips and techniques", in which he introduced the tools and techniques commonly used for load balancing in the cloud. Z Chaczko, V. Mahadevan, S. Aslanzadeh, and C. Mcdermid, in their paper "Availability and load balancing in cloud computing,2013" described the role that load balancing plays in improving the performance and maintainingstability[6][15].

Nishant et al. [7][15] used the ant colony optimization method in nodes load balancing. Randles et al.[9][15] gave a compared analysis of some algorithms in cloud computing by checking the performance time and cost. They concluded that the ESCE algorithm and throttled algorithm are better than the Round Robin algorithm in terms of performance time and cost.

The Round Robin algorithm is the simplest algorithm the uses the concept of time quantum or slices which play a very important role for scheduling, because if time quantum is very large then Round Robin Scheduling Algorithm is same as the FCFS Scheduling. So for simplicity we use the RR algorithm for our work[1][15].

2.1 Cloud Partitioning model:

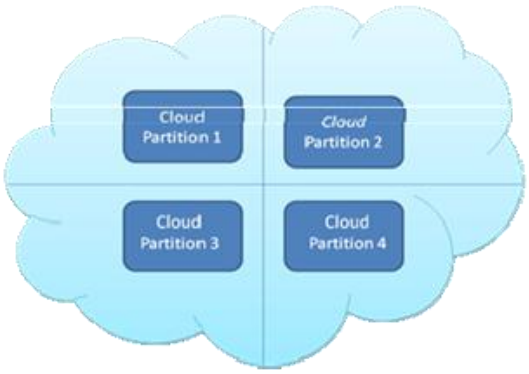

Figure 1: CloudPartitioning

The load balancing strategy is based on the cloud partitioning concept as shown in fig1.After creating the cloud partitions, the load balancing then starts: when a job arrives at the system, then the main balancer decides which cloud partition should receive the job. The partition load balancer then decides how to assign the jobs to the nodes. When the load status of a cloud partition is normal, this partitioning can be accomplished locally. If the cloud partition load status is not normal, this job should be transferred to another partition.

2.2 Main Balancers and Partition Balancers

The load balance solution is done by the main controller and the balancers. The main controller also called the main balancer first receives the incoming jobs and then assigns these jobs to the suitable cloud partition and then communicates with the balancers in each partition to refresh the status information. Since the main controller deals with information for each partition, smaller data sets will lead to the higher processing rates. The balancers in each partition gather the status information from every node and then choose the right strategy to distribute the jobs. The relationship between the main balancer, partition balancers and nodes is shown in Fig.

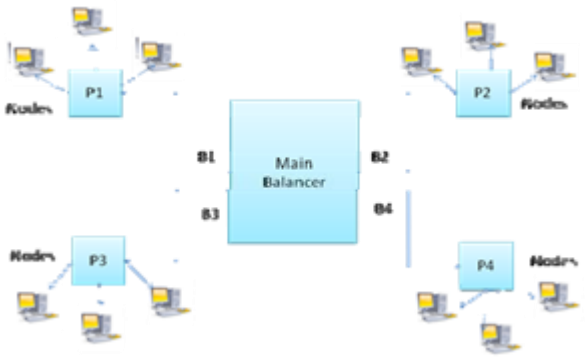

Figure 2: Relationship between the main balancer, partition balancers andnodes

Assigning jobs to the cloud partition. When a job arrives at the public cloud, the first step is to choose the right partition. The cloud partition status can be divided into three types:

1)Idle: When the percentage of idle nodes exceeds α, change to idle status.

2)Normal: When the percentage of the normal nodes exceeds β, change to normal load status.

3)Overload: When the percentage of the overloaded nodes exceeds γ, change to overloaded status.

The parameters α, β, and γ are set by the cloud partition balancers. The main controller has to communicate with the balancers frequently to refresh the status information. The main controller then uses the following strategy to dispatch the jobs: When job i arrives at the system, the main controller queries the cloud partition where job is located. If this location's status is idle or normal, the job is handled locally. If not, another cloud partition is found that is not overloaded.

3. CLOUD PARTITION LOAD BALANCING STRATEGY

1) Load balance strategy for the idle status:

When the cloud partition is idle, many computing resources are available and relatively few jobs are arriving. In this situation, this cloud partition has the ability to process jobs as quickly as possible so a simple load balancing method can be used. There are many simple load balance algorithm methods such as the Random algorithm, the Weighted Round Robin, and the Dynamic Round Robin [15]. The Round Robin algorithm is used here for its simplicity.

The Round Robin algorithm [1] is one of the simplest load balancing algorithms, which passes each new request to the next server in the queue. The algorithm does not record the status of each connection so it has no status information. In the regular Round Robin algorithm, every node has an equal opportunity to be chosen. However, in a public cloud, the configuration and the performance of each node will be not the same; thus, this method may overload some nodes. Thus, an improved Round Robin algorithm is used, which called "Round Robin based on the load degree evaluation" [1].

The algorithm is still fairly simple. Before the Round Robin step, the nodes in the load balancing table are ordered based on the load degree from the lowest to the highest. The system builds a circular queue and walks through the queue again and again. Jobs will then be assigned to nodes with low load degrees. The node order will be changed when the balancer refreshes the Load Status Table.

However, there may be read and write inconsistency at the refresh period T. When the balance table is refreshed, at this moment, if a job arrives at the cloud partition, it will bring the inconsistent problem. The system status will have

changed but the information will still be old. This may lead to an erroneous load strategy choice and an erroneous nodes order. To resolve this problem, two Load Status Tables should be created as: Load Status Table 1 and Load Status Table 2. A flag is also assigned to each table to indicate Read or Write.When the flag = "Read", then the Round Robin based on the load degree evaluation algorithm is using this table. When the flag = "Write", the table is being refreshed, new information is written into this table. Thus, at each moment, one table gives the correct node locations in the queue for the improved Round Robin algorithm, while the other is being prepared with the updated information. Once the data is refreshed, the table flag is changed to "Read" and the other table's flag is changed to "Write". The two tables then alternate to solve the inconsistency.

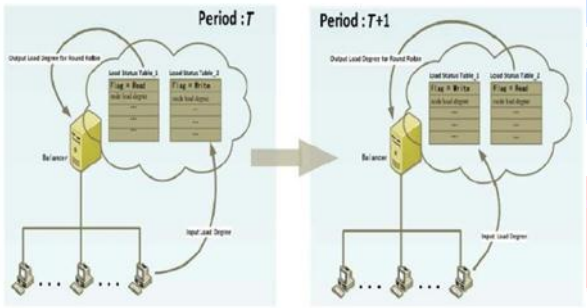

Figure 3: Solution of inconsistencyproblem

2)Load balancing strategy for the normal status:

When the cloud partition is normal, jobs are arriving much faster than in the idle state and the situation is far more complex, so a different strategy is used for the load balancing. Each user wants his jobs completed in the shortest time, so the public cloud needs a method that can complete the jobs of all users with reasonable response time. Penmatsa and Chronopoulos [12] proposed a static load balancing strategy based on random allocation for distributed systems. And this work provides us with a new review of the load balance problem in the cloud environment. As an implementation of distributed system, the load balancing in the cloud computing environment can be viewed as a game.

4. PROPOSED APPROACH

The proposed approach is to design load balancing model for cloud based on partitioning concept with a switch mechanism to choose different strategies for different situations. The load balancing model aimed at the public cloud which has numerous nodes with distributed computing resources in many different geographic locations. Thus, this model divides the public cloud into several cloud partitions. When the environment is very large and complex, these divisions simplify the load balancing. The cloud has a main controller that chooses the suitablepartitionsforarrivingjobswhilethebalancerforeachcloudpartitionchoosesthebest load balancing strategy. There are several cloud computing categories with this work focused on a public cloud. A public cloud is based on the standard cloud computing model, with service provided by a service provider. A large public cloud will include many nodes and the nodes in different geographical locations. Cloud partitioning is used to manage this large cloud. A cloud partition is a subarea of the public cloud with divisions based on the geographic locations. Fig.4 shows proposed architecture and their function is as shown below.

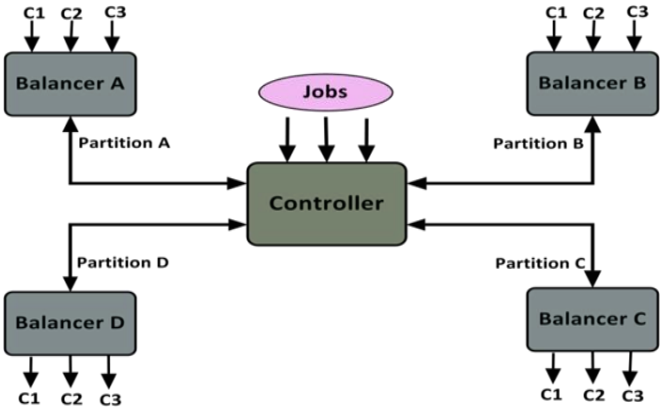

Figure 4: Proposed Architecture

The load balancing strategy is based on the cloud partitioning concept. Once creating the cloud partitions, the load balancing then starts: when a job arrives at the system, with the main controller deciding which cloud partition should receive the job. The partition load balancer then decides how to assign the jobs to the nodes. When the load status of a cloud partition is idle and normal, this partitioning can be accomplished jobs locally. When load status not normal, search another cloud partition. Main aim of load balancing model is improve the efficiency of cloud environment by applying different load balancing algorithms as best partition searching, round robin algorithm, and proposed Random allocation algorithm used in different status. Flowchart shows the overall flow of proposed work in fig 5.

4.1 Advantages of Proposed approach

1.The proposed system is dynamic and there is equally the cloud partition is made to balance the load between n numbers of partition

2.Dynamic round robin algorithm is used in the proposed system in which the system will take less time and less cost to balance the load.

3.When job arrives the cloud partition will start the load balancing to schedule the job in the cloud.

4.Strategy for overloaded servers--add incoming requests in queue and check for server availability after scheduled period [6]

5.Set refresh period for controller and cloud partition balancers to refresh the status at a fixed interval.

Figure 5: Flow chart of proposed work

5. CONCLUSION

Main aim of load balancing model is in order to improve performance and maintain stability of processing so many jobs in public cloud. Load balancing is reduce processing and response time which is having impact on cost. This objective is achieve by constructing good balancing model for public cloud based on cloud partitioning with switch mechanism to choose different strategy to improve the efficiency in public cloudenvironment.

6. REFERENCES

[1]GaochaoXu, Junjie Pang, and Xiaodong Fu, A Load Balancing Model Based on Cloud Partitioning for the Public Cloud, IEEE TRANSACTIONS ON CLOUD COMPUTING YEAR 2013.

[2]Mithra P B, P Mohamed Shameem, "A Novel Load Balancing Model for Overloaded Cloud Computing, IJERT,2012.

[3] M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis, and A.Vakali, Cloud computing: Distributed internet computing for IT and scientific research, Internet Computing, vol.13, no.5, pp.10-13, Sept.-Oct. 2009.

[4] P. Mell and T. Grance, The NIST definition of cloud computing,http://csrc.nist.gov/publications/nistpubs/800 -145/SP800-145.pdf, 2012

[5] N. G. Shivaratri, P. Krueger, and M. Singhal, Load distributing for locally distributed systems, Computer, vol. 25, no. 12, pp. 33-44, Dec. 1992.

[6] B. Adler, Load balancing in the cloud: Tools, tips and techniques, Load-Balancing-in-the-Cloud.pdf, 2012.

[7]Z.Chaczko, V.Mahadevan, S.Aslanzadeh and C. Mcdermid, Availability and load balancing in cloud computing, presented at the 2011 International Conference on Computer and Software Modeling, Singapore, 2011.

[8] K. Nishant, P. Sharma, V. Krishna, C. Gupta, K. P. Singh, N. Nitin, and R. Rastogi, Load balancing of nodes in cloud using ant colony optimization, in Proc. 14th International Conference on Computer Modelling and Simulation (UKSim),Cambridge shire, United Kingdom, Mar. 2012, pp. 28-30.

[9] M. Randles, D. Lamb, and A. Taleb-Bendiab, Acomparative study into distributed load balancing algorithms for cloud computing, in Proc. IEEE 24th International Conference on Advanced Information Networking and Applications, Perth, Australia, 2010, pp. 551-556.

[10]A.Rouse,Publiccloud,http://searchcloudcomputing.techtarget .com/definition/public-cloud, 2012

[11] D. MacVittie,Intro to load balancing for developers: Thealgorithms,https://devcentral.f5.com/blogs/us/introto -load-balancing-for-developers-ndash-thealgorithms,2012.

[12]Peter Mell, Timothy Grance, "The NIST Definition of CloudComputing",http://csrc.nist.gov/publications/nistpubs /800-145/SP800-145.pdf, 2012.

[13] S. Penmatsa and A. T. Chronopoulos, Game-theoretic static load balancing for distributed systems, Journal of Parallel and Distributed Computing, vol. 71, no. 4, pp. 537-555, Apr. 2011.

[14] 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.

[15] 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.

[16]NehaG.Khan,V.B.Bhagat,An Systematic Overview on Cloud Computing and Load Balancing in the Cloud International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 11, Nov - 2013

[17]Tejinder Sharma, Vijay Kumar Banga ,Efficient and Enhanced Algorithm in Cloud Computing,International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-1, March 2013.

[18]Nidhi Jain Kansal,Cloud Load Balancing Techniques: A Step Towards Green Computing, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1, January 2012.

[19]Nidhi Jain Kansal and InderveerChana, Existing load balancing techniques in cloud computing: A systematic re-view, journal of information systems and communication issn: 0976-8742, e-issn: 0976-8750,
volume 3, issue 1, 2012.

[20] Mishra , Ratan , Jaiswal, Anant, P,Ant Colony Optimization: A Solution Of Load Balancing In Cloud, April 2012, International Journal Of Web & Semantic Technology;Apr2012, Vol. 3 Issue 2, P33

[21] Eddy Caron, Luis Rodero-Merino,Auto-Scaling, Load Balancing And Monitoring In Commercial And Open-Source Cloud Research Report ,January2012

[22] Z. Zhang, and X. Zhang, A Load Balancing Mechanism Based on Ant Colony and Complex Network Theory in Open Cloud Computing Federation, Proceedings of 2nd International Conference on Industrial Mechatronics and Automation (ICIMA), Wuhan, China, May 2010, pages 240-243.

[23] Ram Prasad Padhy, P Goutam Prasad Rao,Load Balancing In Cloud Computing Systems, Department of Computer Science and Engineering National Institute of Technology, Rourkela,Orissa, India.pdf.

[24] DoddiniProbhulingL.,Load balancing algorithms in cloudcomputing,International Journal of Advanced Computer and Mathematical Sciences ISSN 2230-9624. Vol4, Issue3, 2013.

INTERNATIONAL JOURNAL OF COMBINED RESEARCH & DEVELOPMENT