International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

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

Neha Gohar Khan, Prof. V. B. Bhagat (Mate)

P. R. Patil College of Engineering & Technology, Amravati, India

Abstract: ***Load balancing is one of the main challenges in cloud computing which is required to distribute the dynamic workload across multiple nodes to ensure that no single node is overwhelmed. Load balancing in the cloud computing environment has an important impact on the performance. Good load balancing makes cloud computing more efficient and responsive. This paper intends to give a better load balance strategy for the public cloud using the cloud partitioning concept. This cloud partitioning would be provided with a switch mechanism for choosing different strategies for different situations. The algorithm applies the game theory to the load balancing strategy to improve the efficiency in the public cloud environment which ultimately helps to improve the different performance parameters like throughput, response time, latency etc. for the clouds.***

Keywords: load balancing model; public cloud; cloud partition; game theory; cloud status.

1. Introduction

Cloud computing services can be used from diverse and widespread resources, rather than remote servers or local machines. There is no standard definition of Cloud computing. 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 specific role[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 underloaded[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 the system[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 special features[19].

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,2011" described the role that load balancing plays in improving the performance and maintaining stability[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 that 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

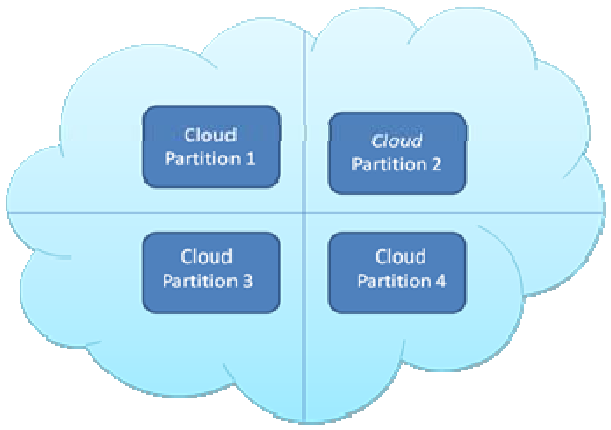


Figure 1: Cloud Partitioning

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. This is all about the concept. Now let's see how it work step by step.

2.2 Main balancer 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.

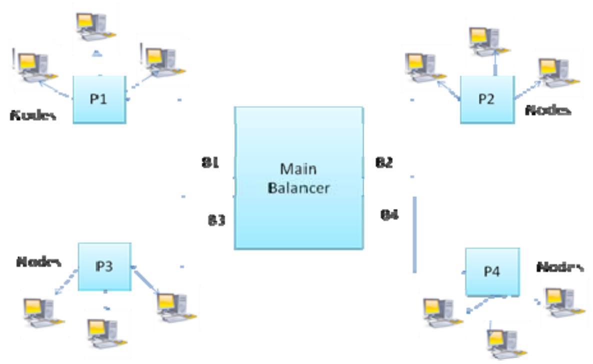


Figure 2: Relationship between the main balancer, partition balancers and nodes

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.

Assigning jobs to the nodes in the cloud partition

The cloud partition balancer gathers load information from every node to evaluate the cloud partition status. This evaluation of each node's load status is very important. The first task is to define the load degree of each node. The node load degree is related to various static parameters and dynamic parameters. The static parameters include the number of CPU's, the CPU processing speeds, the memory size, etc. Dynamic parameters are the memory utilization ratio, the CPU utilization ratio, the network bandwidth, etc. The load degree is computed from these parameters as below:

Step 1: Define a load parameter set: F= {F_1; F_2; _ _ _ ; F_m) with each F_i ($1\leq i \leq m$, F_i [0,1]))parameter being either static or dynamic. m represents the total number of the parameters.

Step 2: Compute the load degree as:

$Load\_degree\ (N) = \sum_{i=1}^{m} \alpha_i F_i$,

$\alpha_i(\sum_{i=1}^{m} \alpha_i = 1)$ are weights that may differ for different kinds of jobs N represents the current node.

Step 3: Define evaluation benchmarks. Calculate the average cloud partition degree from the node load degree statistics as:

$$Load\_degree_{avg} = \frac{\sum_{i=1}^{n} Load\_degree(N_i)}{n}$$

The bench mark Load_degree $_{high}$ is then set for different situations based on the Load degree$_{\_avg}$.

Step 4: Three nodes load status levels are then defined as:

- Idle :- When

$Load\ \_degree\ (N) = 0$

there is no job being processed by this node so the status is charged to Idle.

- Normal :- For

$0 < Load\ \_degree\ (N) \leq Load\_degree_{high}$

the node is normal and it can process other jobs.

- Overloaded :- When

$Load\_degree_{high} \leq Load\ \_degree\ (N)$

the node is not available and cannot receive jobs until it returns to the normal.

The load degree results are input into the Load Status Tables created by the cloud partition balancers. Each balancer has a Load Status Table and refreshes it each fixed period T. The table is then used by the balancers to calculate the partition status. Each partition status has a different load balancing solution. When a job arrives at a cloud partition, the balancer assigns the job to the nodes based on its current load strategy. This strategy is changed by the balancers as the cloud partition status changes.

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.

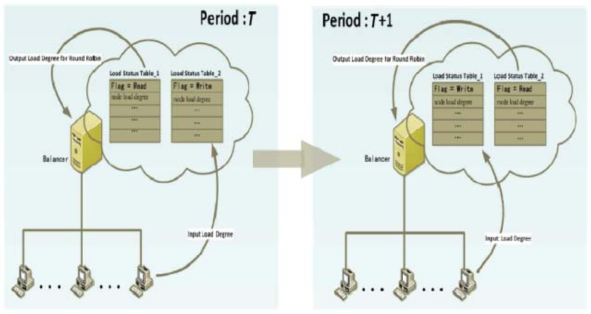


Figure 3: Solution of inconsistency problem

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 game theory 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.

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 others. In non-cooperative games, each decision maker makes decisions only for his own benefit. The system then reaches the Nash equilibrium, where each decision maker makes the optimized decision. The Nash equilibrium is when each player in the game has chosen a strategy and no player can benefit by changing his or her strategy while the other player’s strategies remain unchanged.

There have been many studies in using game theory for the load balancing. Grosu et al. [13] proposed a load balancing strategy based on game theory for the distributed systems as a non-cooperative game using the distributed structure. They compared this algorithm with other traditional methods to show that their algorithm was less complexity with better performance. Aote and Kharat[14] gave a dynamic load balancing model based on game theory. This model is related on the dynamic load status of the system with the users being the decision makers in a non-cooperative game. Since the grid computing and cloud computing environments are also distributed system, these algorithms can also be used in grid computing and cloud computing environments. Previous studies have shown that the load balancing strategy for a cloud partition in the normal load status can be viewed as a non-cooperative game, as described here.

The players in the game are the nodes and the jobs. Suppose there are n nodes in the current cloud partition with N jobs arriving, then we define the following parameters:

μ_i :: Processing ability of each node, i=1, _ _ _, n.

ϕ_j : Time spending of each job.

$\phi = \sum_{j=1}^{N} \phi_j$: Time spent by the entire cloud partition,

$\phi < \sum_{i=1}^{n} \mu_i$

s_{ji} : Fraction of job j that assigned to node i($\sum_{i=1}^{n} s_{ji}$ =1 and $0 \le s_{ji} \le 1$).

4. Proposed Work

A. The proposed work is divided into the following modules:

1. User Module

In this module, Users will be having authentication and security to access the details which is presented in the ontology system. Before accessing or searching the details user should have the account in the system otherwise they have to get register first by filling the registration form.

2. System Module

This system module is focused on a public cloud which is based on the standard cloud computing model, with services provided by a service provider. The system model is based on the concept that a large public cloud will include many nodes and the nodes in different geographical locations. Cloud partitioning will be used here to manage this large cloud 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 normal, this partitioning can be accomplished locally. If the cloud partition load status is not normal, this job should be transferred to another partition.

Moreover as the number of users grows on in the cloud, the system model will provide ways to divide the public cloud into more locations as clouds servers. Also the system model will show the location of the current servers with its location with its location number, location name and location description. The system model also provides available server in a particular location which keeps the information of servers as server name, server URL, total connections, current connections and status as ideal, normal or overloaded. The system model will graphically shows users that logged-in on a particular location with its IP address i.e. clients IP and also shows the total number of users on that particular location.
In short, system model will provide a way to add new locations, new servers and monitor all the servers on all locations for current connections and current status.

b. Main Controller and Balancers

The load balance solution is done by the main balancer and the partitioned balancers. The main controller first assigns jobs to the suitable cloud partition and then communicates with the balancers in each partition to refresh this status information. The balancers in each partition gather the status information from every node and then choose the right strategy to distribute the jobs.

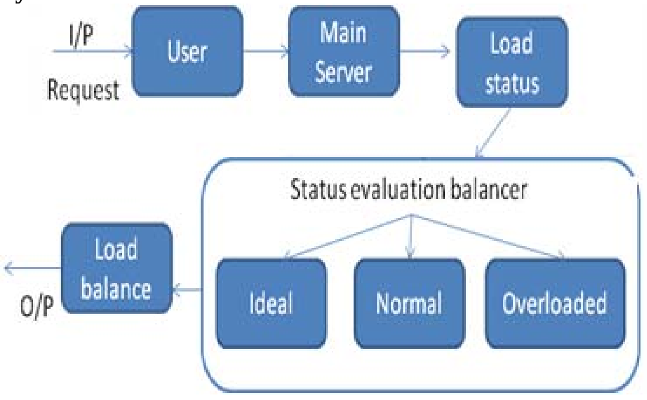
User will request from main cloud. Then main controller will redirect the user to a corresponding location. If this corresponding location is overloaded, the main controller will redirect the user to the other nearby location. After redirecting it to the appropriate location that locations main server will dynamically check the status of all available servers under that location generally called as partitioned balancers. Now this balancer will handle the user's job uptil the user's logout. The user's login and logout to one of the available balancers from all the locations will update the main controller with current status.

5. Cloud Partition Load Balancing Strategy

When the cloud partition is idle, few jobs are arriving and thus the cloud partition has the ability to process jobs as quickly as possible so a simple load balancing method such as "The Round Robin algorithm based on the load degree evaluation" will be used here for its simplicity.

When the cloud partition is normal, job arrival pattern much faster than in the idle state and the situation is far more complex, so a different strategy is used for the load balancing as each user wants his jobs completed in the shortest time. The current model uses the game theory approach for non-cooperative games proposed by Grosu[1][13] called "the best reply" to calculate sji of each node, with a greedy algorithm used to calculate sji for all nodes.

System Model



b. Objectives:
The objectives to be achieved from the proposed model are…
1. Enhance the performance of the cloud.
2. Reduce traffic in the cloud.
3. Increasing reliability.
4. Increasing throughput.

6. Conclusion & Future Scope

Effective performance enhancement is achieved with the help of load balancing in cloud computing environment. Proper load balancing will leads to efficient cloud computing and higher user satisfaction. Paper proposed the new architecture for load balancing as well as performance enhancement in public cloud. The concept of load balancing is implemented on the basis of cloud partitioning method with different strategies for different cloud status. Our primary aim is to achieve the results and compare those with the existing system. Significant level of development is running to satisfy the required objectives.

References

[1] Gaochao Xu, 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] R. Hunter, *The why of cloud*, http://www.gartner.com/DisplayDocument?doccd=226469&ref= g noreg, 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,

[12] 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] Neha G.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 Inderveer Chana, *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 Optimiza tion: 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] Doddini Probhuling L.,*Load balancing algorithms in cloudcomputing*,International Journal of Advanced Computer and Mathematical Sciences ISSN 2230-9624. Vol4, Issue3, 2013.

[25] Microsoft Academic Research, Cloud computing, http://libra.msra.cn/Keyword/6051/cloud-computing?query= cloud%20computing, 2012.

[26] Google Trends, Cloud computing, http://www.google.com/trends/explore#q=cloud%20computing, 2012.

[27] http://www.livinginternet.com/w/wionline.htm.

[28] www.cloudbus.org/cloudsim.

[29] http://www.ca.com/~/media/Files/whitepapers/turnkey_clouds_turnkey_profits.pdf.