International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Implementation of Cloud Partitioning based Load Balancing for Performance Improvement

Neha Gohar Khan[1], V. B. Bhagat (Mate)[2]

[1]P .R. Patil College of Engg & Technology, Amravati, Maharashtra, India

[2]Professor, P .R. Patil College of Engg & Technology, Amravati, Maharashtra, India

Abstract: ***Cloud computing has been widely adopted by the industry due to its ease of use and simple service oriented model. The number of users accessing the cloud services keeps on increasing day-by-day. In such a situation, load balancing is one of the complex challenges in cloud computing which is required to distribute the dynamic workload across multiple nodes because prediction of user request arrivals on the server is not possible. Load balancing in the cloud computing environment has an important impact on the performance. Good load balancing makes cloud computing more efficient and responsive. Cloud partitioning is an optimal approach in public cloud for load balancing. In this paper, we are trying to implement a better load balance strategy for the public cloud using the cloud partitioning concept to improve the performance in the public cloud environment.***

Keywords: Cloud computing, load balancing, cloud partitioning, dynamic round robin

1. Introduction

Cloud computing is a fast growing area in computing research and industry today. Cloud computing is a computing paradigm, where a large pool of systems are connected in private or public networks, to provide dynamically scalable infrastructure for application, data and file storage. It uses internet and central remote servers to maintain data and application. A Cloud system consists of three major components such as clients, datacenter, and distributed servers [3][12].

Cloud computing provide infrastructure, platform, and software as services. These services are using pay-as-you-use model to customers, regardless of their location. Cloud computing is a cost effective model for provisioning services and it makes IT management easier and more responsive to the Changing needs of the business. Today, network bandwidth, less response time, minimum delay in data transfer and minimum data transfer cost are main challenging issues in cloud computing load balancing environment [4][7].

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 under loaded [17]. 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 [20].

With the advancement of the Cloud, there are new possibilities opening up on how applications can be built and how different services can be offered to the end user through virtualization, on the internet. There are the cloud service providers who provide large scaled computing infrastructure defined on usage, and provide the infrastructure services in a very flexible manner which the users can scale up or down at their own will. The establishment of an effective load balancing algorithm and how to use cloud computing resources efficiently for effective and efficient cloud computing is one of the cloud computing service provider's ultimate goals [15][19].

In this paper, firstly we are tried to propose a better load balance strategy for the public cloud using the cloud partitioning concept which could help to improve the performance of the cloud. Secondly, a dynamic load balancing algorithm has been implemented for an IaaS framework in virtual cloud computing environment.

Limitations of Existing System:

• Static schemes do not use the system information and cannot be applicable in real time.

• Only one main server is present to handle all the coming requests. If it gets overloaded due to large number of requests and if the request is coming for the same cloud server, the request will have to wait for response from the server until the server load gets minimized, which will cause latency problem.

Different research papers were studied related to load balancing in the cloud where numerous proposed load balancing algorithms have been compared on the basis of their advantages. But every research paper has some limitation as there is no single dynamic strategy which can deal with different load balancing situations at runtime and more work need to be done to reduce the response time of jobs. So, we will try to overcome some of the limitations of existing system.

2. Proposed Work

The load balancing model given in this article is aimed at the public cloud which has numerous nodes with distributed computing resources in many different geographic locations. Users request from all over the globe in huge numbers simultaneously; so it becomes very difficult to manage such a large cloud. For simplicity, this work divides the public cloud into several cloud partitions. Thus, the load balancing strategy is based on the cloud partitioning concept. After creating the cloud partitions, the load balancing starts: when a job arrives at the system, then the main balancer decides which cloud partition should receive the job. Then partition cloud and the main cloud both execute the incoming jobs.

3. Implementation

To handle the random selection based load distributed problem, we have used a dynamic scheduling algorithm i.e. dynamic round robin and estimated response time and processing time, which is having an impact on performance.
We are using the following techniques:
1. Cloud Partitioning
2. Dynamic Round-Robin Technique for load balancing

Flow diagram of the system

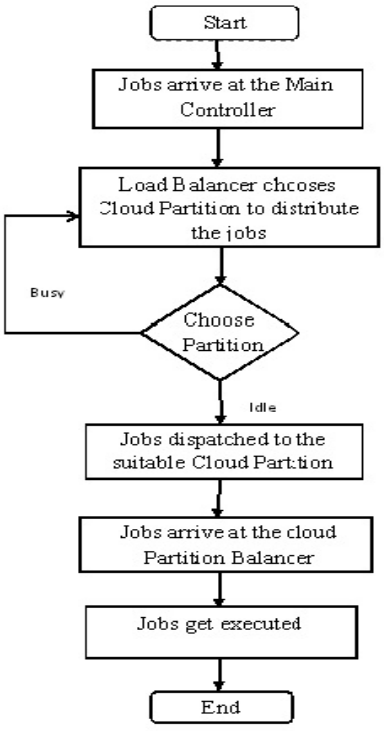

Figure 1: Flow diagram

Modules Description: Our proposed work is divided into the following modules:

1. User Module

In this module, users will be having authentication and security to access details which are presented in the ontology system. The users can perform a number of activities during their visit to the cloud. 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. Admin Module

This module is focused on load balancing. As the number of users grows on the server, the model will provide ways to divide the public cloud into more locations as clouds servers. Admin can add or edit a particular server with its id, name, and location name and location description and most IP address. The load balancing module also provides load information under load tables which keeps the information of servers as server name, server URL, current load on the servers i.e. number of jobs and status as ideal, normal or busy. In short, this module provides a way to add new servers, provide current status of each server as idle, normal and busy, number of jobs in the queue, and the job response time and execution time. In GUI for system, there is a dialog box for admin to access the server.

3. File sender and Receiver

This module is focused on file sending and receiving. It works with the help of RMI technology in java. It is used to continuously send the file to the server from the client side in order to generate sufficient load on the server to balance it among the partition servers.

Generalized System Architecture

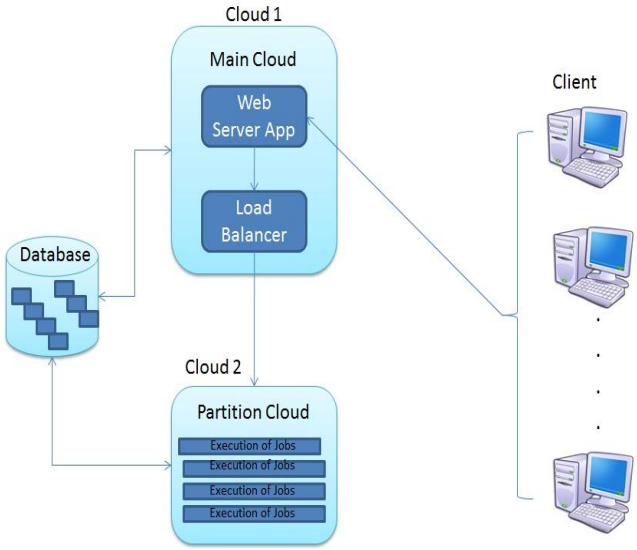

Figure : System Architecture

I. Dynamic Round Robin algorithm-1 for main cloud

Step 1: Start
Step 2: Read: Jobs arrive at the main controller
Step 3: Receive the jobs
Step 4: Execute the jobs
Step 5: Repeat
Step 6: End

II. Dynamic Round Robin algorithm-2 for main cloud and partition cloud

Step 1: Start
Step 2: Read: Jobs arrive at the main controller
Step 3: Receive the jobs
Step 4: Divide the jobs by N number of servers
Step 5: Schedule the first part for the first server
Step 6: Schedule the second part for the second server

Step 7: Schedule the last part on the N^{th} server
Step 8: Execution of jobs
Step 9: Repeat
Step 10: End

Steps for status evaluation
Three server load status levels are defined as:
- Idle :- When Load _degree (N) > 0
- Normal :- For0 <Load _degree (N) ≤ $Load\_degree_{high}$
- Busy :- When $Load\_degree_{high}$ ≤ Load _degree(N)

Where load_degree limit is set by the load balancer

4. Result Analysis

As part of the implementation work, two virtual clouds are created on separate systems on the basis of IP addresses sharing common database and application. The main controller i.e. cloud1 receives these jobs and sends them to the load balancer which splits the jobs and distributes it among the main cloud and the partition cloud for execution. According to the partition load, the status of partition is calculated as idle, normal or overloaded. The execution time and the response time were calculated at both the servers for the incoming jobs. Also, a separate batch of jobs was sent to the main server for comparing the results of cloud partitioning method and the execution time and the response time were calculated separately.

Table 1: Server performance result table

Connected servers	Total jobs	Response Time (in millisec)	Execution Time (in millisec)
Server 1	30	52	33
Server 2	30	20	32
Main server	60	26	64

Table 2: Server Status table

Server	Total jobs	Jobs executed	Jobs remaining	Server status
Server1	30	05	25	Busy
Server2	30	27	03	Idle

The data, thus collected was used to plot a runtime graph consisting of number of jobs, response time and execution time of the jobs for the two partitioned clouds as well as for the single cloud.

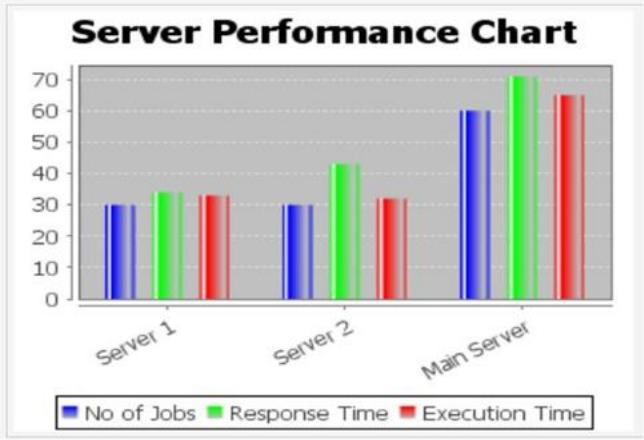

Figure: Server Performance Graph

It was estimated that the single server became very busy with the execution of all the queries whereas with partitioning method, the server did not get overloaded with continuous load and also it did not crash. Also, better response time and execution times were obtained in partitioning method. Thus, the overall estimated performance of the system was better with cloud partitioning as compared to execution of jobs on a single cloud for our simulation model.

5. Conclusion & Future Scope

Cloud Computing is a vast and popular concept in the modern age and load balancing plays a very important role in case of Clouds. Cloud partitioning is a method to make partitions of huge public cloud in some segment of cloud. There is a huge scope of improvement in this area. This model demonstrated the applicability of using cloud partitioning method and then using Dynamic Round Robin algorithm for load balancing to obtain measurable improvements in resource utilization and availability of cloud-computing environment and increase the business performance in cloud based sector. According to the partition load, status of partition is calculated. Other load balancing strategy can give better results and improve the performance.

So, tests are needed to compare different load balancing strategies. Many tests are needed to guarantee system availability and efficiency. A better framework will be needed for cloud division methodology.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] Nidhi Jain Kansal and Inderveer Chana, *Existing load balancing techniques in cloud computing: A systematic re-view*, journal of information systems and

[16] communication issn: 0976-8742, e-issn: 0976-8750, volume 3, issue 1, 2012.

[17] 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

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

[19] 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.

[20] Doddini Probhuling L.,*Load balancing algorithms in cloud computing*,International Journal of Advanced Computer and Mathematical Sciences ISSN 2230-9624. Vol4, Issue3, 2013.

[21] Neha G.Khan,V.B.Bhagat,Cloud Partitioning Based Load Balancing Model For Performance Enhancement In Public Cloud, International Journal of Science and Research (IJSR),Volume 3 Issue 9, September 2014.