MJRET
Open Access
Multidisciplinary Journal of Research in Engineering and Technology
Journal homepage: www.mjret.in
ISSN:2348 - 6953

LOAD BALANCING STRATEGY BASED ON CLOUD PARTITIONING CONCEPT

Ms. Shilpa D.More[1], Prof. Arti Mohanpurkar[2]
[1,2]Department of computer Engineering
DYPSOET, Pune,India
shilpa.jaware@gmail.com,yasharti@gmail.com

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 random allocation for 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 for the clouds.*

Keywords*: load balancing model; public cloud; cloud partitioning; cloud computing;*

1. INTRODUCTION

Cloud computing is an attracting technology in the field of computer science. In Gartner's report, it says that the cloud will bring changes to the IT industry. The cloud is changing our life by providing users with new types of services. Users get service from a cloud without paying attention to the details. [7]NIST gave a definition of cloud computing as a model for enabling ubiquitous, 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. More and more people pay attention toward cloud computing.

Cloud computing is efficient and scalable but maintaining the stability of processing so many jobs or requests in the cloud computing environment is a very complex problem .So the job arrival pattern is not predictable and the capacities of each node in the cloud differ, for load balancing problem, load control is crucial to improve system performance and maintain stability. The load balancing model is aimed at the public cloud which has numerous nodes with distributed computing resources in many different geographic locations. 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. RELATED WORK

There have been many studies of load balancing for the cloud environment. Cloud computing is a recent trending in IT that moves computing and data away from desktop and portable PCs into large data centers. Load balancing in cloud computing was described in a white paper written by Adler[2] who introduced the tools and techniques commonly used for IEEE TRANSACTIONS ON CLOUD COMPUTING YEAR 2013 load balancing in the cloud. However, load balancing in cloud is still a new problem that needs new architectures to adapt too many changes. Chacko et al.[3] described the role that load balancing plays to improve the performance and maintaining stability.

There are many load balancing techniques given by the researchers over time to time same have advantages over and vice-versa. Distribute load of multiple network links to achieve maximum throughput, minimize response time and to avoid overloading. There are many loads balancing algorithm such as Round Robin algorithm, Throttled algorithm, Equally Spread Current Execution Algorithm, Ant Colony algorithm. Randles et.al.[4]giving the comparative analysis by checking cost and performance.

3. 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 suitable partitions for arriving jobs while the balancer for each cloud partition chooses the

best 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.1 shows proposed architecture and their function is as shown below.

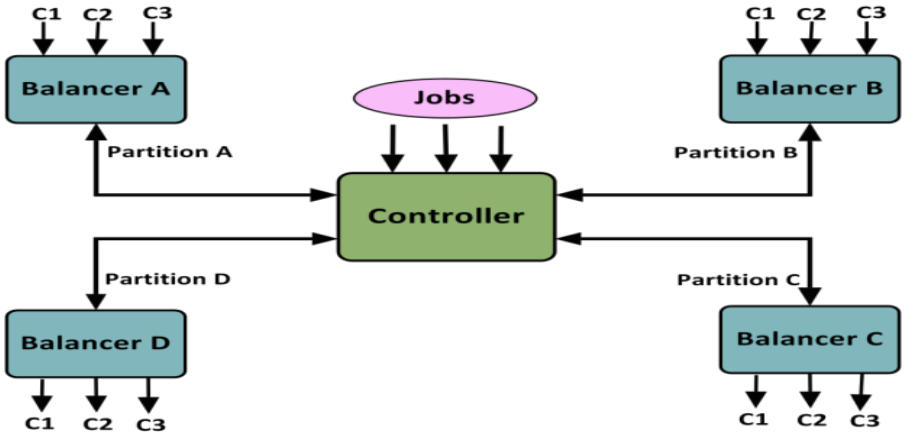

Fig.1: Proposed Architecture

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. Flow chart shows overall flow of proposed work in Fig.2

3.1 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

Fig.2: Flow chart of proposed work

4. WHAT IS LOAD BALANCING

Load balancing is the technology to distribute workload across multiple computers or a computer cluster, central processing units, disk drives, RAM, or other resources, to achieve optimal resource utilization, maximize throughput, minimize response time, avoid overload, and minimize application down time. The load balancing services is usually provided by dedicated software or hardware.

Purpose of load balancing includes:

- To distribute the load amongst a number of machines/connections
- To provide redundancy in case one machine/server fails
- Effective distribution of traffic among multiple servers
- Works as a driver rather than as a service
- Manages resources efficiently
- Improves the application response time

5. IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. The implementation, designing is methods to achieve change over and evaluation of change over methods.

5.1 SYSTEM MODEL

The proposed system is based on cloud partitioning concept. The load balance solution is done by the main controller and the balancers. The main controller first assigns jobs to the suitable cloud partition and then communicates with the balancers in each partition to refresh status information. Then 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 or connection and then choose the right strategy to distribute the jobs to node.

Good load balance will improve the performance of the entire cloud. Therefore, the current model integrates several methods and switches strategy for the load balancing based on the system status. Here, the idle status uses Round Robin algorithm while the normal status uses random allocation algorithm based load balancing strategy.

5.2 CLOUD PARTITION LOAD BALANCING STRATEGY

When a job arrives at the public cloud, the first step is to choose the right partition. The main controller has to communicate with the balancers frequently to refresh the status information. The main controller then dispatches the jobs using the following strategy: When n no. job 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. When request/job is from same or shortest server then job sending that same server using best partition searching algorithm is shown in below

5.3 BEST PARTITION SEARCHING ALGORITHM

Begin
while job do
search Best Partition (job);
if partition State == idle // partition State == normal then
Send Job to Partition;
else
search for another Partition;
end if

end while

end

Three different load status levels are defined as:

- Node status is Idle, there is no load being processed. Use Round Robin algorithm.
- Node status is Normal, then node is process some other load. Use Random allocation algorithm.
- Node status is overloaded, then node is not available and not receive job. Add incoming request in queue and check for server availability after schedule period.

5.4 LOAD BALANCING STRATEGY FOR 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. First, the nodes in the load balancing table are ordered based on the load from the lowest to the highest and two Load Status tables. Load balancing strategy for the idle node status uses Round robin based on bandwidth availability. A system builds a circular queue and walk through queue again and again. Then assign job to connection or node using high bandwidth.

5.5 LOAD BALANCING STRATEGY FOR 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 .In order to achieve this based on random allocation algorithm.

This algorithm based on randomly selecting one condition out of three

If(r==0), Allocation will happen based on distance algorithm (based on latitude/longitude)

If(r==1), Allocation will happen based on available bandwidth

If(r==2), Allocation will happen based on lowest load

6. EXPERIMENTAL RESULT AND ANALYSIS

In this phase, n numbers of request will send for load balancing using different strategy under different conditions as idle, normal and overloaded stage. Our request is sending same or shortest server using best partition searching algorithm, if another partition is idle status use round robin algorithm and when normal status uses proposed random allocation

algorithm. Fig.3 shows numbers of request/load sending to cloud partition then do their allocation using different strategy with respective execution time (millisecond). Fig 4 shows comparison of Best partition Searching, Round robin and Random allocation Algorithms.

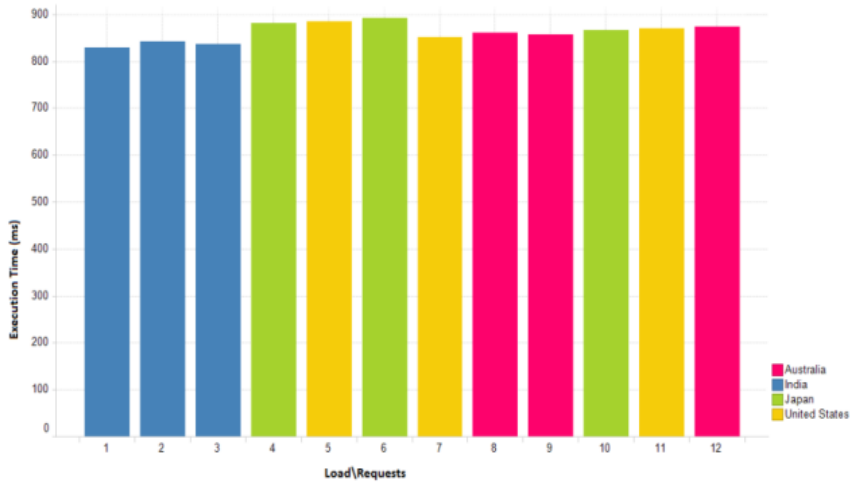

Fig.3: Analysis load v/s Execution time

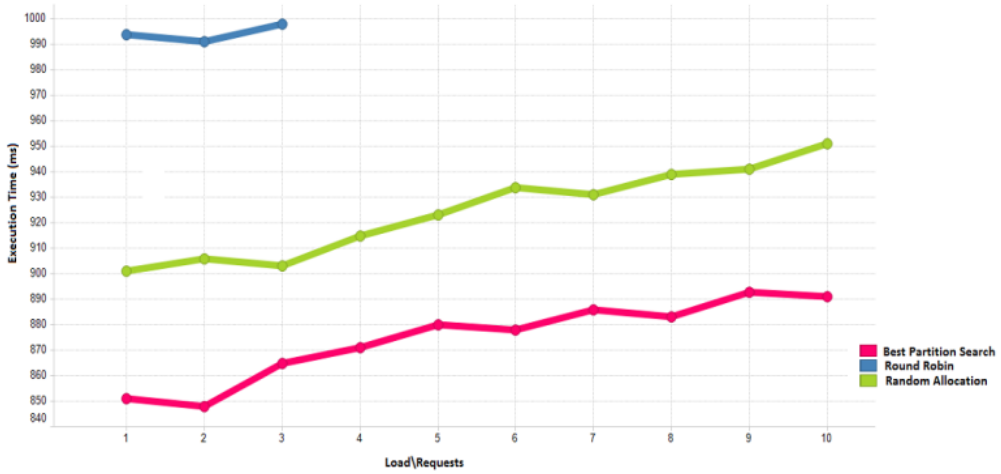

Fig.4: Comparison of Algorithms

7. 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 cloud environment.

ACKNOWLEDGEMENT

The authors would like to thank to Gaochao Xu, JunjiePang, and Xiaodong Fu for their exclusive information and valuable comments.

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] *B. Adler, Load balancing in the cloud: Tools, tips and techniques, http://www.rightscale.com/info center/white Papers/Load-Balancing-in-the-Cloud, pdf, 2012*

[3] *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*

[4] *Zhong Xu, Rong Huang, (2009),"Performance Study of Load Balancing Algorithm in Distributed Web Server System",CS213 Parallel and distributed system*

[5] *Ms.NITIKA, Ms.SHAVETA, Mr. GAURAV RAJ; "Comparative Analysis of Load Balancing Algorithm in Cloud Computing" International Journal of Advance Research in Computer Engineering & Technology Volume 1,2012*

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

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

[8] *D. MacVittie, Intro to load balancing for Developers-The algorithms, https://devcentralf5.com/blogs/us/introto- Load -balancing-for-Developers-ndash-the-algorithms, 2012*