ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Journal Impact Factor: 2.114

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

AN EFFICIENT FUZZY LOAD BALANCING ALGORITHM FOR PUBLIC CLOUDS

Anindita Kundu
Assistant Professor, Dept. of Information Technology, Sadabai Raisoni Women's College, Nagpur, Maharashtra, India.

ABSTRACT

Cloud computing is efficient and scalable but maintaining the stability of processing so many jobs in the cloud computing environment is a very complex problem with load balancing receiving much attention for researchers.. Cloud computing means distributed computing. Cloud computing enables convenient, on-demand, dynamic and reliable use of distributed computing resources. Load balancing in the cloud computing environment has an important impact on the performance. It makes cloud computing more efficient and improves user satisfaction. This article introduces a better load balance model for the public cloud. To improve the efficiency in the public cloud environment, this algorithm is being introduced that applies fuzzy inference with load scheduling.

KEYWORDS: Cloud computing, Fuzzy inference, Load Scheduling, Public cloud

INTRODUCTION

Cloud computing has become the rapidly growing area in industry today with the advancements in the field of science and technology. Cloud computing by using the resources, information, software and shared equipment provides a client's service within a specific time. Cloud computing is a way to increase the capacity or add capabilities dynamically without investing in new infrastructure, training new personnel, or licensing new software. It extends Information Technology's (IT) existing capabilities. There are cloud service providers who provide large scale computing infrastructures as services in which the users can use it according to their requirements. In present day scenario of the network cascaded with task consummation, and the aspect of heterogeneity along with platform divergence, dynamic load balancing plays a vital role in optimizing the performance of the server in the cloud computing environment [1].

The cloud computing model has five main characteristics:

- on-demand service
- broad network access
- resource pooling
- flexibility
- Measured service.

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 [2].

The load measuring is applied so as to avoid disruption in delivery of a service when one or more components of the system are in trouble. Load balancing comes up d when a system fails to respond, and does not sent traffic on it. Often, problems can be minimized with proper load balancing that not only reduce costs and create green computing.

LITERATURE SURVEY

Load balancing in cloud computing was described in a white paper written by Adler who introduced the tools and techniques commonly used for load balancing in the cloud. However, load balancing in the cloud is still a new problem that needs new architectures to adapt to many changes [2]. The load balancing algorithm is done before it reaches the processing servers the job is scheduled based on various parameters like processor speed, memory utilization and assigned load of Virtual Machine and etc [3]. The ultimate goal of load balancing is as follows:

- Even distribution of load to each resource
- Minimization of processing time for each job
- Maximum utilization of each resource

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Journal Impact Factor: 2.114

Any algorithm concerning load balancing is designed on the basis of static and dynamic. Static algorithms do not depend upon the current state of Algorithms face a major drawback in case of sudden failure of system resource and tasks. Dynamic algorithms take decisions concerning load balancing based upon the current state of the system and don't need any prior knowledge about the system [4]. This approach is an improvement over the static approach. The algorithms in this category are considered complex, but have better fault tolerance and overall performance. There are often many dynamic load balancing algorithms, such as Round Robin, Honeybee Foraging, Active Clustering, Biased Random Sampling, Equally Spread Current Execution Algorithm, and many more.

Challenges for Load Balancing

There are some qualitative metrics that can be improved for better load balancing in cloud computing.

Throughput: It is the total number of tasks that have completed execution for a given scale of time. It is required to have high through put for better performance of the system.

Associated Overhead: It describes the amount of overhead during the implementation of the load balancing algorithm. It is a composition of movement of tasks, inter process communication and inter processor. For load balancing technique to work properly, minimum overhead should be there.

Fault tolerant: We can define it as the ability to perform load balancing by the appropriate algorithm without arbitrary link or node failure. Every load balancing algorithm should have good fault tolerance approach.

Migration time: It is the amount of time for a process to be transferred from one system node to another node for execution. For better performance of the system this time should be always less.

Response time: In Distributed system, it is the time taken by a particular load balancing technique to respond. This time should be minimized for better performance.

Resource Utilization: It is the parameter which gives the information within which extant the resource is utilized. For efficient load balancing in system, optimum resource should be utilized.

Scalability: It is the ability of load balancing algorithm for a system with any finite number of processor and machines. This parameter can be improved for better system performance.

Performance: It is the overall efficiency of the system. If all the parameters are improved then the overall system performance can be improved.

the system and have prior knowledge regarding system resources.

SYSTEM MODEL

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. The cloud partitioning strategy proposed at [3] where the load balancer partitions the nodes into a number of sub-areas based on the load state, and each partition is thereby assigned a load state value. The main controller applies a best partition search algorithm to identify the partition and transfers the job to the respective load balancer [8]. The architecture is shown in Fig 1. The load balancing strategy is based on the cloud partitioning concept.

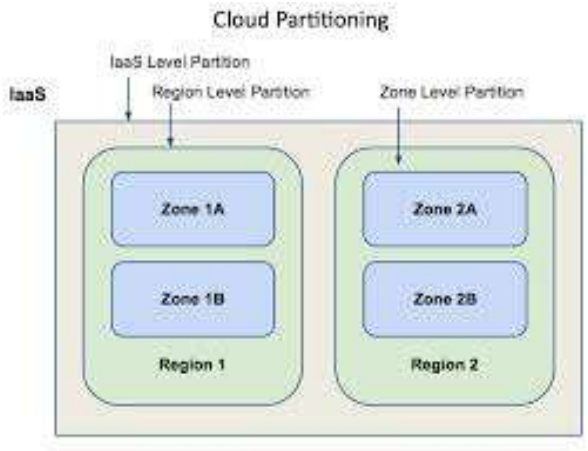

Fig.1. Typical Cloud Partitioning

For partitioning the job, main controller and balancer is required. The difference between these two is shown in fig. 2.

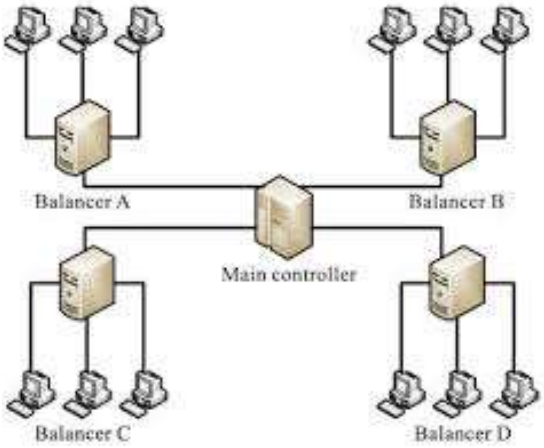

Fig. 2. Main controller distribute the load to the balancer.

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Journal Impact Factor: 2.114

Load balancing policy in the general cloud is as follows:
When jobs arrive at the system, 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 balancer checks the status information from every node and then chooses appropriate partitions and nodes based on fuzzy inference to distribute the jobs. If the cloud partition load status is not normal, then the job should transfer to another partition. The whole process is shown in Fig 3.

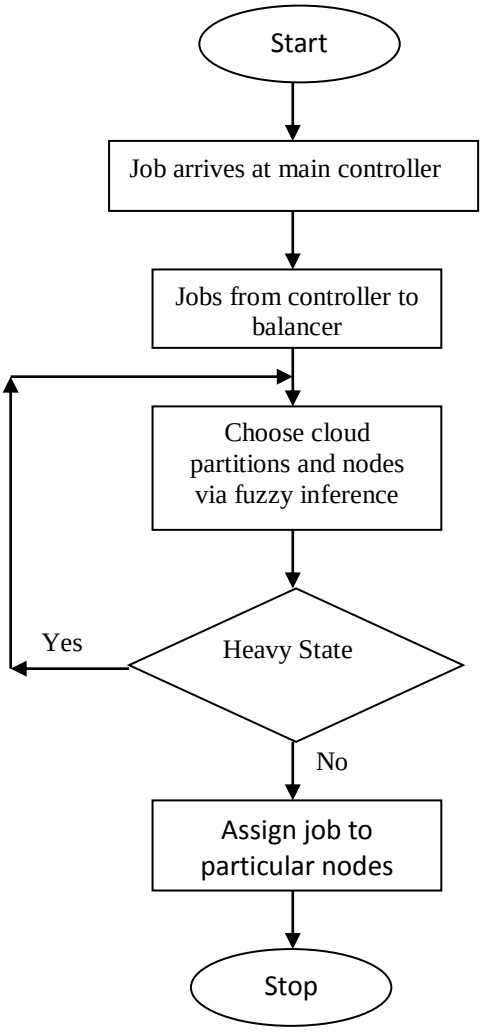

Fig.3. Job Assignment Strategy

CLOUD PARTITION STRATEGY

As in fig. 3 above we see that, there are two states, normal state and idle state. A relatively simple method can be used for the partition of idle state. The load balancers then switch methods as the status changes. 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. 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 [6].

EVALUATION NODES

The first task is to determine the appropriateness of the nodes. The appropriateness of nodes 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.

Nodes are evaluated in three steps.

Step1.Define Parameter Set: It is assumed that there are two parameters for evaluating the appropriate node:

- Memory Utilization
- Time Delay

Step2. Determining the appropriateness of nodes in the system:
Assume that the system on which a job is to be performed is comprised of K partitions (j=1,...K).

Assume that the jth partition (j=1,...K) in job allocation system includes nj nodes. Then the appropriateness of the rth parameter from the ith node of the jth partition is indicated as α_{rij}. which is defined as follows [5].

$$\alpha_{rij}=\frac{F_{rij}}{\sum_{j=1}^{k}\sum_{i=1}^{n}F_{rij}} \in (0,1) r = 1,2$$

F_{rij} is numeral value assigned to the rth parameter from the ith node of the jth partition. For instance, if the rth parameter is time delay, will be the distance between job orderer location and the ith node from the jth partition, where r=1, 2 are the two parameters discussed as above.

Step3.Choose nodes via fuzzy inference:
Phase 1.Fuzzification
Fuzzification in the appropriateness of parameters:

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Journal Impact Factor: 2.114

The linguistic variable used to represent the node there are three levels to represent the node memory utilization: low, medium and high, respectively and there are three levels to represent the node time delay: close, adequate and far, respectively. The membership functions developed and their corresponding linguistic states are represented in tables1 through 2 and Fig.4 through 5. Response time in cloud computing was calculated in [7]

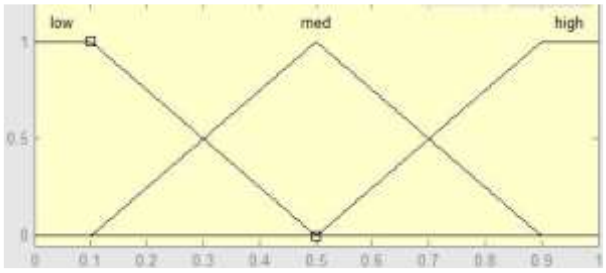

Fig. 4. Fuzzy set for fuzzy variable $\alpha_{(Memory\ Utilization)}$

Table 1: Ranges of Memory Utilization

Fuzzy Linguistic term	Parameters
Low	[0.0 0.1 0.5]
Medium	[0.1 0.5 0.9]
High	[0.5 0.9 1.1]

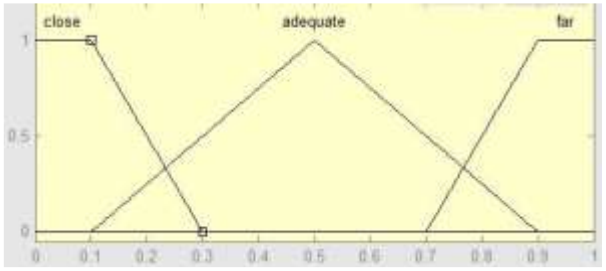

Fig. 5. Fuzzy set for fuzzy variable $\alpha_{(Time\ Delay)}$

Table 2: Ranges of Time delay

Fuzzy Linguistic term	Parameters
Close	[0.0 0.1 0.3]
Adequate	[0.1 0.5 0.9]
Far	[0.7 0.9 1.1]

The two inputs named as Memory utilization and time delay gives a single output, i.e., the node.
It is shown in fig. 6. The different linguistic variables for the node presentation are: excellent, good, rather good, medium, rather bad, bad and very bad.

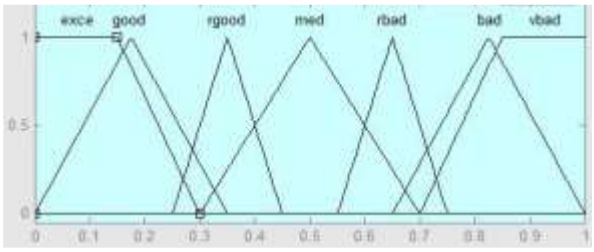

Fig.6. Fuzzy set for fuzzy output- mode

Table 3: Ranges of output

Fuzzy Linguistic term	Parameters
Excellent	[0.0 0.15 0.3]
Good	[0.0 1.75 0.35]
Rather good	[0.25 0.35 0.45]
Medium	[0.3 0.5 0.7]
Rather bad	[0.55 0.65 0.75]
Bad	[0.65 0.825 1]
Very bad	[0.7 0.85 1.1]

Table 3 represents the range of outputs.

Phase 2: Fuzzy Rule Base
The fuzzy rule base currently includes rules like the following:
1. If (Memory utilization is low) and (Delay time is close) then (output is excellent).
2. If (Memory utilization is low) and (Delay time is adequate) then (output is good).
3. If (Memory utilization is low) and (Delay time is far) then (output is good).
4. If (Memory utilization is medium) and (Delay time is close) then (output is good).
5. If (Memory utilization is medium) and (Delay time is adequate) then (output is medium).
6. If (Memory utilization is medium) and (Delay time is far) then (output is rather bad).
7. If (Memory utilization is high) and (Delay time is close) then (output is rather bad).
8. If (Memory utilization is high) and (Delay time is adequate) then (output is bad).
9. If (Memory utilization is high) and (Delay time is far) then (output is very bad).

CONCLUSION

Load balancing is one of the main challenges in Cloud Computing. It is required to distribute the load evenly at every system to achieve a high user satisfaction and resource utilization. In this paper, I introduced the load balancing algorithm using fuzzy logic in cloud computing, in which load balancing is a core and challenging issue in Cloud Computing. This article introduced a better load balance model for the public cloud based on the cloud partitioning concept with a switch mechanism to choose different strategies for

different situations. In this way, the load of the data cloud can be reduced more accurately and the performance of heavily loaded cloud storage system can be greatly improved by using this system. Cloud network has a large extent of application in today's industry. It is used in various fields such as industrial development, entertainment, medical etc. hence; the challenges too increase aside. Hence, this paper has a greater scope of future work.

REFERENCES

1. A Fuzzy-based Firefly Algorithm for Dynamic Load Balancing in Cloud Computing Environment, N. Susila, S. Chandramathi, Rohit Kishore, JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE, VOL. 6, NO. 4, NOVEMBER 2014.
2. Analysis of Load Balanced Model for Public Cloud Based on Cloud Partition Method, K. Venkateswarlu1, Y.Leela krishna2 , International Journal of Advanced and Innovative Research (2278-7844) / # 143 / Volume 3 Issue 8.
3. Effective load balancing in cloud computing, Zeinab Goudarzi*, Ahmad Faraahi, International Journal of Intelligent Information Systems 2014; 3(6-1): 1-9 Published online September 26, 2014 (http://www.sciencepublishinggroup.com/j/ijiis).
4. Comparative Study on Load Balancing Techniques in Cloud Computing, N. S. Raghava* and Deepti Singh, OPEN JOURNAL OF MOBILE COMPUTING AND CLOUD COMPUTING Volume 1, Number 1, August 2014.
5. The Fuzzy Load Balancing for hetergeneous Nodes in Public Cloud, Rogheyeh Salehi1, Mohammad Adabitabar Firoozja2, Visi Jurnal Akademik.
6. A Load Balancing Model Based on Cloud Partitioning for the Public Cloud Gaochao Xu, Junjie Pang, and Xiaodong Fu, TSINGHUA SCIENCE AND TECHNOLOGY ISSNl l1007-0214l l04/12l lpp34-39 Volume 18, Number 1, February 2013.
7. S.Mohapatra, S.Mohanty, K.Smruti Rekha, 2009, Analysis of Different Variants in Round Robin Algorithms for Load Balancing in Cloud Computing, International Journal of Computer Applications (0975 – 8887), Volume 69– No.22.
8. Gaochao Xu, Junjie Pang, Xiaodong Fu, "A load balancing model based on cloud partitioning for the public cloud", TSINGHUA SCIENCE AND TECHNOLOGY- ISSNl l1007-0214l l04/12l lpp34-39,Volume 18, Number 1, February 2013

Anindita Kundu is an Assistant Professor of Sadabai Raisoni Women's College, Nagpur, affiliated to SNDT Mumbai University. She is currently pursuing PHD in the topic of Cloud Computing. She has presented 4 national and international conferences and one workshop of Cloud Computing. She is a Counsellor of IGNOU. Also she is university exam paper setter as well as moderator. Her current research interest includes Cloud Computing and Fuzzy Logic.