International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161

Available at http://inpressco.com/category/ijcet

Review Article

Dynamic Load Distribution and Balancing using Cloud Partitioning

Snehal D. Sonawane[Å*] and R. H.Borhade[Å]

[Å]Department of Information Technology, University of Pune, State Maharashtra, Country India

Accepted 10 August 2014, Available online 25 Aug 2014, **Vol.4, No.4 (Aug 2014)**

Abstract

Cloud computing is the emerging and transformational paradigm in the field of information technology. It mostly focuses in providing various services on demand and resource allocation and secure data storage are some of them. To store huge amount of data and accessing data from such metadata is new challenge. Distributing and balancing of the load over a cloud using cloud partitioning can ease the situation. Implementing load balancing by considering static as well as dynamic parameters can improve the performance cloud service provider and can improve the user satisfaction. Implementation the model can provide dynamic way of resource selection depending upon different situation of cloud environment at the time of accessing cloud provisions based on cloud partitioning.

Keywords: *Cloud partitioning, Load balancing model, dynamic parameter.*

1. Introduction

Data is mainly stored in distributed manner in case of cloud computing. Data is further stored in remote locations. If it is going to be uploaded randomly on cloud it would result in imbalance of storage of cloud server. For instance if some of nodes are heavily loaded while others negligible relating to very less load i.e. 10 Gb data loaded on one server and 0 GB data on other server.

Cloud computing is a technology that aims to provide services on internet on scalable basis to organizations via Cloud vendors. It gives ability of utilizing services to organization without installation and maintenance problem.

Data is secured in cloud and will be accessed by only authorised personnel's, who are specified by cloud service provider. Hence for such cases, access control system is used.

From different geographical locations, many nodes would be included in a large public cloud. To manage large cloud partitioning concept is used. There is a subarea of public cloud with respect to different geographic locations based on divisions.

Main controller decides after creating cloud partitions that which cloud partition should receive the job. Job assigned to a node is decided by partition load balancer.

To accomplish partition locally load status of a cloud partition should be normal.

For cloud technology many researchers involved for load balancing. Alder introduced the technique of load balancing in cloud computing. Different tools and techniques were introduced by him that is commonly used for load balancing. New architecture needs to be implemented and various changes need to be adapted as load balancing in cloud is still a new difficulty. Load balancing plays important role in improving performance as well as maintaining stability.

2. Related Works

Many studies are conducted regarding cloud computing regarding load balancing, comprising various load balancing algorithms. Few of them are Round Robin, Equally Spread Current Execution Algorithm, and Ant Colony algorithm. Some efficient algorithm also includes Nishant ET. Ant colony optimization method is used in node load balancing. Cost and performance time by comparison of analysis of other algorithms are checked by Randles ET algorithm .Better results are provided by ESCE algorithm rather than Round Robin algorithm. Allocation method used in operating system and few traditional load balancing methods are same. For instance Round Robin algorithm and the First Come First Served (FCFS) algorithm are similar but it is simplicity which allows us to use Round Robin algorithm.

Constantly improving ratio of Cost vs. performance in networks of workstations has been noticed. It will be continued in near future. For fastest parallel computers aggregate peak rate matches or exceeds peak rate of such computers. However a serious difficulty emerges in concurrent programming that is to deal with scheduling and load balancing for such systems that comprises of heterogeneous computers.

Different service users will get advantage in flexibility, cost and availability by anticipated use of Cloud computing that is built on well-established research work in utility computing networks, web services, distributed computing and virtualization. Further demand for cloud computing would rise relating to earlier benefits increasing

*Corresponding author: **Snehal D. Sonawan**

Cloud's customer base and various Cloud installations scale.

In various Service Oriented Architectures and Internet of services (IOS) - type applications various technical issues are implicated such as scalability, fault tolerance and high availability. Load balancing technique is central to all such issues. Centralized assignment of jobs to particular servers is made infeasible by complexity and scale of these systems, hence requiring an effective distributed solution.

Data is stored in distributed way in Cloud computing. Remote locations or virtual locations are main destinations to save data. There is imbalance in cloud server storage if data is loaded randomly. For instance if some of nodes are heavily loaded while others negligible relating to very less load i.e. 10 Gb data loaded on one server and 0 GB data on other server.

3. Proposed Work

Efficient technique for load balancing in cloud is proposed by load balancing model which is based on cloud partitioning for the public cloud. Different geographic locations have a public cloud with various nodes and distributed computing resources. Hence public Cloud could be divided into several cloud partitions.

These divisions can help in load balancing whenever there is huge and complex environment. Hence for arriving jobs required partitions can be chosen by main controller. The best possible load balancing strategy is chosen by balancer of each cloud partition.

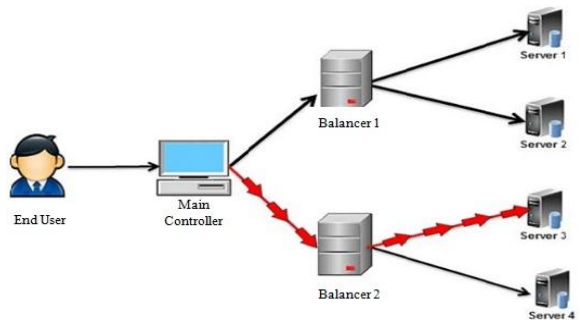

Fig. 1: Implementation Scenario

Different geographic locations have a public cloud with various nodes and normal; job should be transferred to different partition distributed computing resources. Hence public Cloud could be divided into several cloud partitions.

Load balancing starts at the job arrival, during creation of cloud partition is carried out, while the main controller decides which Cloud partition would be receiving the job individually. Jobs assigned to a particular node are decided by partition of load balancer. Partitioning is accomplished locally when cloud partition is normal for a load status. Further if, cloud partition load status is

Application Servers:

- Cloud architecture may contain one to many number of application servers according to its scope and utility.
- Each application server has its number of dedicated resources

Master Servers:

- Master server is the first component which interacts with client and accept its request
- It divide task into number of activities

A load balancing model based on cloud partitioning for the public cloud proposed efficient technique for balancing load in cloud.

Define a load parameter set: F = {F1; F2;…; Fm} with each Fi($1 \leq i \leq m$; Fi $\in$ [0, 1]) parameter being considered. m represents the total number of the parameters.

Then Compute the load degree as:

Load_degree(N) = $\sum_{i=1}^{m}$ aiFi

a_i are weights that may differ for different kinds of jobs

Load $\text{degree}_{avg} = \frac{\sum_{i=1}^{m} Load\_Degree(Ni)}{n}$

Step 1 :

Get size of all Servers S1…..Sn

$S = \sum_{i=1}^{s} aiSi$

Step 2:

When, if S ∈ Server load(sL) is overflowed (limit exceeds) then

File fu uploads on S ∈ server next to sL.

Where

1) Load is Idle When

Load degree.(N) = 0;

2) Load is Normal when

$0 <$ Load degree (N) $\leq$ $\text{Load degree}_{high}$

3) Load is Overload when

Load_degree high $\leq$ Load_degree(N)

$$Si = \sum_{n=1}^{n} fbi + Pi$$

Where S_i is the server of respective Balancer, fb_i is the file size of each server and P_i are the processes of each server respectively which will be considered dynamically.

Assume that, Balancer 1 has minimum load

e.g. Load_Balancer1 < Load_Balancer2 then Load_Balancer1 should be considered and selected for the service provision.

4. Implementation Details

Whenever the environment is complex and huge, the load balancing can be simplified by these divisions. Then the he suitable partitions can be chosen by a main controller for arriving jobs. However, the balancer of each cloud partition chooses the best suitable load balancing strategy.

In the currently proposed system the size of the server has not been set and also it is not taken as a parameter while uploading the load to the server so there might be a

risk of overflowing the server. To avoid this, following contribution will help in building efficient system.

It may possible that one of the servers exceeds its size beyond the limit and also runs out of available bandwidth. The application will check the available space and actual size of the server. It may vary from each other and also the load of the server it contains so. After uploading the file it will check the size of file and the file will be uploaded to the other server if its size is greater than the available size. In this way, one can prevent Overflow of the server.

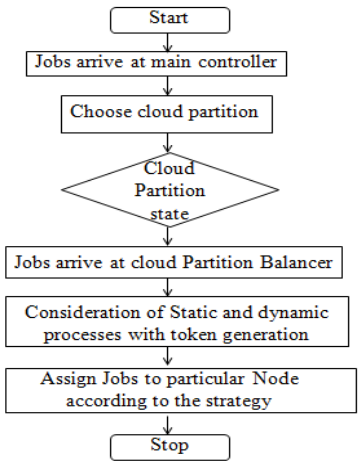

Fig. 2: Job assignment strategy

Upload time Formula

In calculating the uploading time(in sec) of File we are considering the bandwidth, file size in KB, and no. of servers.

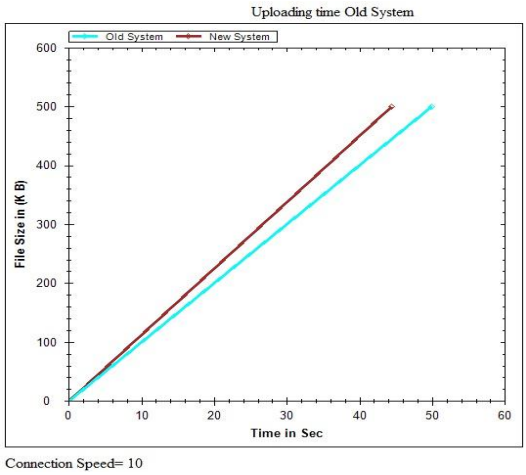

Fig. 3: Graph for uploading a file

Existing system:

Time (T) = FileSize/ConnectionSpeed

e.g T = 500/10 = 50sec (approx)

Proposed System:

Time (T) = FileSize/ConnectionSpeed –(P/nS *1/nS)

Here, P is the minimum processes running on the server and nS is the number of servers i.e Probability.

e.g T= 500/10 – (90/4 * 1/4) = 44.38sec (approx)

(Note: Times are approximate. Connections slowdowns may affect actual Upload time.)

Download time Formula

In calculating the uploading time (in sec) of File we are considering the bandwidth, file size in KB, and no. of servers. As in existing system the first servers are overloaded as compared to the next ones. So automatically it will effect on their response time which will be more.

Time (T) = RET – RST

RET= Response End Time

RST= Request Start Time;

```
i=Counter
j=index(id) of file
RST
{
  for(i=0;i<=no.of file;i++)
  {
    If(j==fileid[i])
    Response End Time;
  }
  Response End Time;
}
```

In existing system, as there are n….no. of files as compared to new system, so the time taken for existing system will be greater than the new system, as it is distributive and efficient.

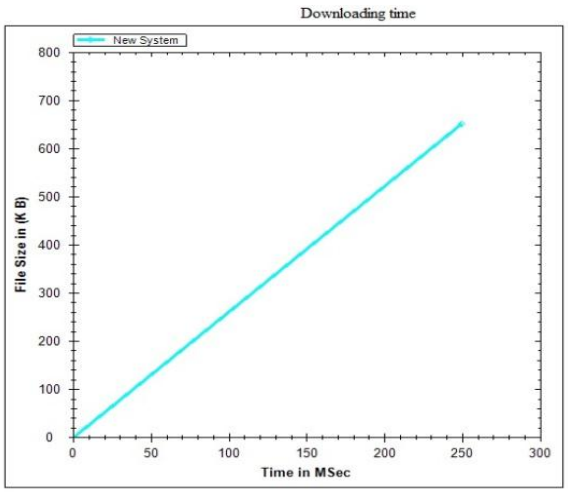

Fig. 4: Graph for downloading a file

Thus, the linear graph that shows the comparison of the present system and projected system grounded on the count of the observations necessary for load balancing.

Conclusions and Future Scope

The considered system implementation scenario of the cloud involves the client interaction with the cloud service provider taking main controller and balancers into the picture. Cloud service provider partitions the cloud as balancers and the various servers under each balancer. Making a connection of main controller to servers, involves the action of getting load of each balancer.

In implementation scenario, client uploads the files on cloud by using algorithm of load balancer. For this, it selects the balancer of cloud service provider having the minimum load. Each balancer will have the records of server load which will be computed by considering static as well as dynamic parameters. Client can search and download the file and deployments from the server on which file has been uploaded at the time of server selection.

Considering static and dynamic parameters can support the load balancing based on cloud partitioning to improve the efficiency. Involving dynamic parameters takes the current running processes into the pictures. Computation of static and dynamic parameters for balancing results into distributing the load appropriate and effective.

There can be the applications which can balance the load or reduce the operator's cost of service delivery are also feasible, but for this further investigations is needed to ensure operational stability, efficiency and overhead control. In light of the emerging efforts towards providing more security to the system, the latter application can become more pressing.

References

Gaochao Xu, Junjie Pang, and Xiaodong Fu (2013), A Load Balancing Model Based on Cloud Partitioning for the Public Cloud, *Proc. 14th European Conf. Research in Computer Security (ESORICS '09 IEEE TRANSACTIONS ON CLOUD COMPUTING).*

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.

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.

Z. Chaczko, V. Mahadevan, S. Aslanzadeh, and C. Mcdermid (2011), Availability and load balancing in cloud computing, *The 2011 International Conference on Computer and Software Modeling, Singapore.*

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.

M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis, and A. Vakali (2009), Cloud computing: Distributed internet computing, *IT and scientific research, Internet Computing*, vol.13, no.5, pp.10-13.

Stavros Papadopoulos, Spiridon Bakiras, and Dimitris Papadias (2010), pCloud: A Distributed System for Practical PIR, supported by Grant HKUST 618108 from Hong Kong RGC, and by the NSF Career Award IIS- 0845262.

K. Ren, C. Wang, and Q. Wang (2012), Security Challenges for the Public Cloud, *IEEE Internet Computing*, vol. 16, no. 1, pp. 69-73.

A. Rahumed, H. C. H. Chen, Y. Tang, P. P. C. Lee, and J. C. S. Lui., A Secure Cloud Backup System with Assured Deletion and Version Control.,*3rd International Workshop on Security in Cloud Computing*, 2011.