IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 3, Ver. 1 (May – Jun. 2015), PP 13-16
www.iosrjournals.org

Load Rebalancing for Distributed Hash Tables in Cloud Computing

Eng. Ahmed Hassan A/Elmutaal algonsol.me86@gmial.com ,
Dr. Amin Babiker A/Nabi Mustafa amin31766@gmail.com

Abstract:*In cloud computing applications are provided and managed by the cloud server and data is also stored remotely in cloud configuration. As Cloud Computing is growing rapidly and clients are demanding more services and better results, load balancing for the Cloud has become a very interesting and important research area. Load balancing ensures that all the processor in the system or every node in the network does approximately the equal amount of work at any instant of time. In this paper, a fully distributed load rebalancing algorithm is presented to cope with the load balance problem. Our algorithm is compared against a centralized approach in a production system and a competing distributed solution presented in the literature.*

I. Introduction

In cloud computing, load balancing is required to distribute the dynamic local workload evenly across all the nodes. It helps to achieve a high user satisfaction and resource utilization ratio by ensuring an efficient and fair allocation of every computing resource. Proper load balancing aids in minimizing resource consumption, implementing fail-over, enabling scalability, avoiding bottlenecks and over provisioning. There are mainlytwotypesofloadbalancingalgorithms: In staticalgorithmthetrafficdividedevenly amongtheservers.This algorithmrequiresapriorknowledge of systemresources ,sothatthe decision of shiftingoftheloaddoes notdependonthecurrentstateofsystem .Static algorithms roper in thesystemwhichhaslowvariation in load. In dynamic algorithm the lightest server in the whole network or system is searched and preferred for balancing a load. For this real time communication with network is needed which can increase the traffic in the system. Here current state of the system is used to make decisions to manage the load. Load balancing based on Cloud Partitioning There are several cloud computing services with this work focused on a public cloud. A public cloud is based on the standard cloud computing model, with service provided byaservice 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.

EXISTING SYSTEM: However, recent experience concludes that when the number of Storage nodes, the number of files and the number of accesses to files increase linearly, the central nodes become a performance bottleneck, as they are unable to accommodate a large number of file accesses due to clients and Map Reduce applications. Thus, depending on the central nodes to tackle the load imbalance problem exacerbate their heavy loads. Even with the latest development in distributed file systems, the central nodes may still be overloaded.

PROPOSED SYSTEM: In this paper, we are interested in studying the load rebalancing problem in distributed file systems specialized for large-scale, dynamic and data-intensive clouds. (The terms “rebalance” and “balance” is interchangeable in this paper.) Such a large-scale cloud has hundreds or thousands of nodes (and may reach tens of thousands in the future). Our objective is to allocate the chunks of files as uniformly as possible among the nodes such that no node manages an excessive number of chunks.

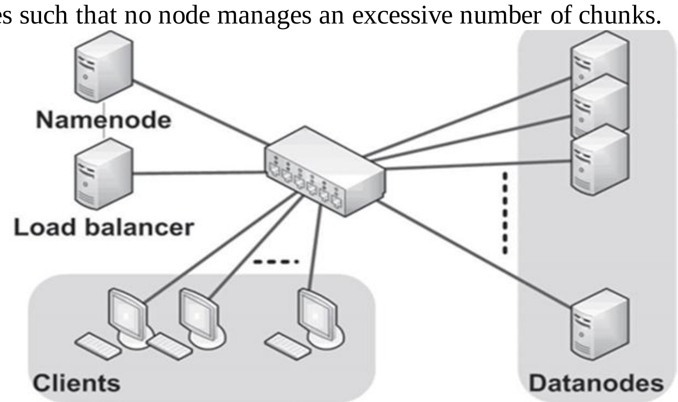

Figure (1): system architecture

II. Methodology

<u>Methodology:</u>

There are many nodes in a public cloud which are at different locations. The cloud has a main controller (**MC**) which chooses the suitable partitions for arriving jobs. The appropriate partition is selected by using best load balancing strategy. All the status information is gathered and analyzed by main controller and balancers. They also perform the load balancing operations. The system status then provides a basis for choosing the right load balancing strategy. In this paper we will use approximately 4 different servers, which are partitioned into small clouds called balancers (each balancer will have some servers). Cloud Service Provider (**CSP**) is used to handle a Main cloud (which is made up of small Clouds) called Main Controller or Controller main. Client interacts with cloud using a web application called client Site.

1. <u>Uploads File:</u>

When client upload files it will be stored in the server. The cloud will take care that it will be loaded into the server which has minimum load. The status of every server is updated by the balancers and depending on the status the partition is selected.

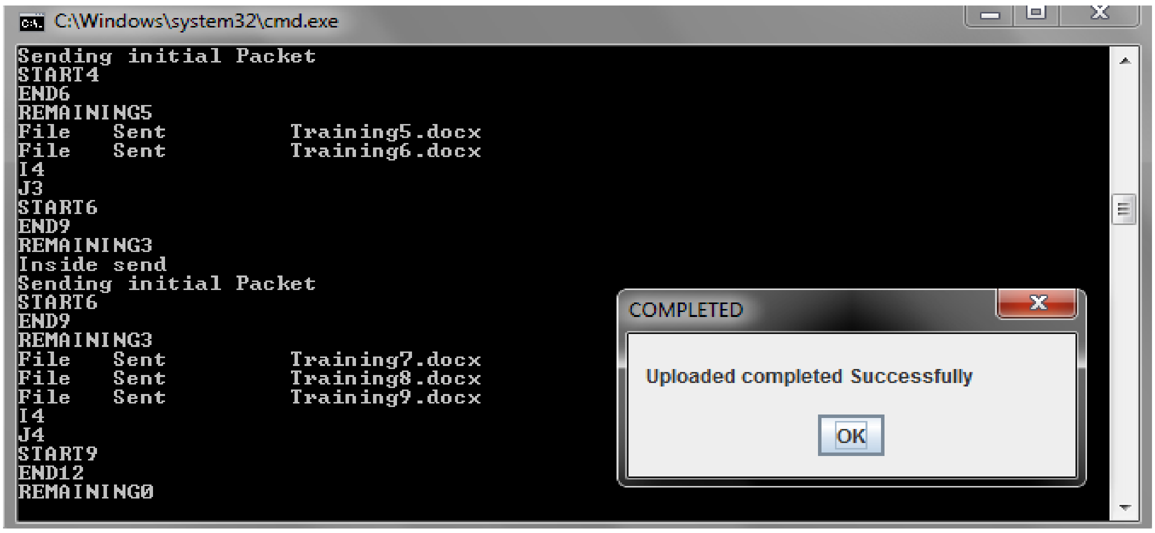

Figure (2): Upload file

2. Download File:

The servers will have following states for user download file:**Idle, Normal, Overloaded**. For overloaded condition another partition is searched.

Partition status can be divided into three types:

(1) **Idle**: When the load exceeds **alpha**
(2) **Normal**: When the load exceeds **beta**
(3) **Overload**: When the load exceeds **gamma**

The parameters alpha, beta, and gamma are set by the cloud partition balancers.
Define a load parameter set : F = {F1 ;F2;…….;Fm} with each Fi(1 ≤i ≤m;Fi€ [0,1])
parameter being either static or dynamic . m represents the total number of the parameter .
Then compute the load degree as :
Load degree (N) = $\sum_{i=1}^{m}=1\alpha_i f_i$
Calculate the average cloud partition degree from the node load degree statistics as:
Load degree $_{avg}$= = $\sum_{i=1}^{m}$ =load degree (N_i)
N
Where :

1) Load is **Idle** when : load degree (N) = 0;
2) Load is **Normal**when : 0< load degree (N) ≤ load degree$_{high}$;
3) Load is **Overload**when : load degree $_{high}$≤ load degree (N)

3. System Configuration:-

We use this flowing server with (s\w & h\w) as billow:-

H/W System Configuration:-

Processor	**Core i3**
Speed	**2.0 GHz**
RAM	**2.0GB (min)**
Hard Disk	**120 GB**
Key Board	**Standard**
Mouse	**Standard**
Monitor	**Standard**

S/W System Configuration:-

Operating System	**Windows 7**
Application Server	**Tomcat 8.0 x**
Front End	**HTML, JAVA, JSP**
Scripts	**JavaScript**
Server side Script	**Java Server Pages**
Database	**My sql**
Database Connectivity	**JDBC**

4. Best Partition Searching Algorithm:

The first step to download files from cloud computing by client is to send a request to (MC) which located in balancer server. Scantly, the balancer server chooses a cloud partition state, where by download speed state has three states **(Idle, Normal and overload)**. In case of **overload** state, the balancer returns the request to recheck for anew partition, however in case of the other two states **(Idle and Normal)** the Jobs arrive at the cloud partition balancer and then Assign jobs to particular nodes according to the strategy in order to complete the download files.

Below are data flow charts that describe the process:

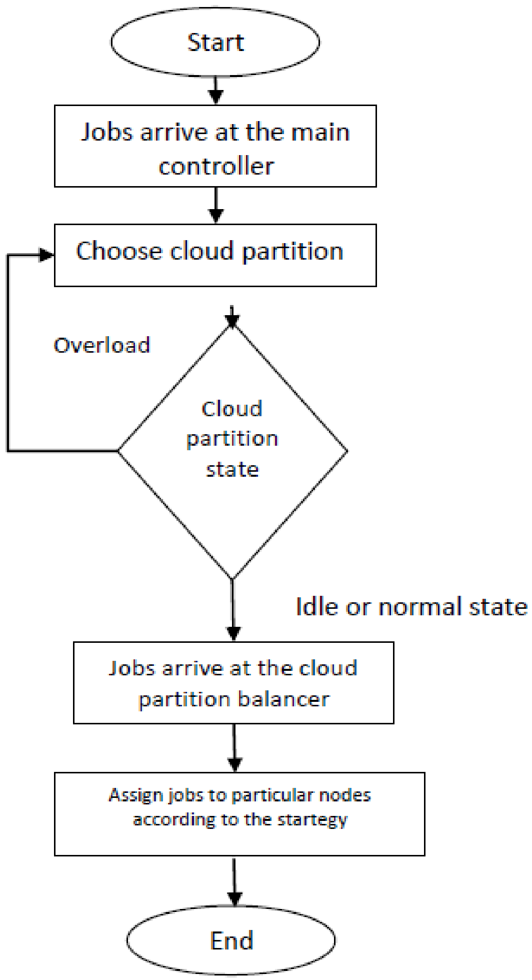

Figure (3): Process flow chart

III. Result:

After implementing a small cloud system by using the two opting (with balancer server & without balancer server) the result was as follows:-

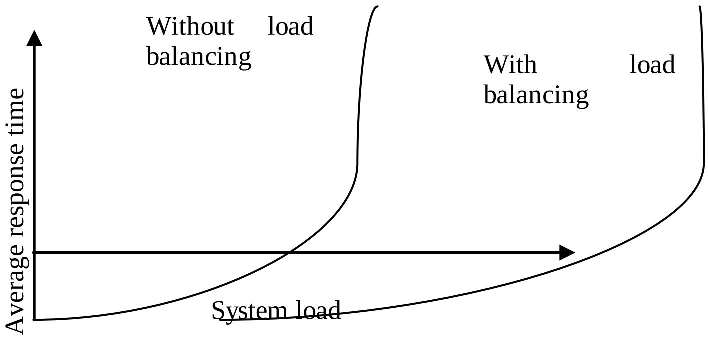

Figure (4): Unbalanced VS Balanced Systems

The Above Mention curve is the result of applying the equations of the load degree by its different values (alpha, beta, and gamma) .the average response time reached the peak when the balancer server been used , additionally the download speedwas very good , less load , more capacity been acquired and the QOS (quality of service) has improved which eventually increases the satisfaction level of the client.

Having said the above, without balancer server option the result was hasn't client satisfaction.

IV. Conclusion:

In line with the objective of this paper which is to balancer load on cloud , experimentally the uses of balancer server has improve the performance of the cloud computing system significantly compere with (without using balancer server) , hence we do recommend to use the balancer server in all cloud systems for biter QOS.

References

[1]. Xu, Gaochao, Junjie Pang, and Xiaodong Fu. "A load balancing model based on cloud partitioning for the public cloud." IEEE Tsinghua Science and Technology, Vol. 18, no. 1, pp. 34-39, 2013.

[2]. P. Mell and T. Grance, ―The NIST definition of cloud Computing‖, online available at: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf, 2012.

[3]. N. G. Shivaratri, P. Krueger, and M. Singhal, ―Load distributing for locally distributed systems‖, Journal Computer, vol. 25, no. 12, pp. 33-44, Dec. 1992.

[4]. Zhu, Yan, Huaixi Wang, Zexing Hu, Gail-JoonAhn, Hongxin Hu, and Stephen S. Yau. "Efficient provable data possession for hybrid clouds." In Proceedings of the 17th ACM conference on Computer and communications security, pp. 756-758. ACM, 2010.

[5]. A. Rouse, ―Public cloud‖, available at: http://searchcloudcomputing.techtarget.com/definition/public-cloud.

[6]. Lori MacVittie, ―Cloud Balancing: The Evolution of Global Server Load Balancing‖, F5 white paper, online available at: http://www.f5.com/pdf/white-papers/cloud-balancing-white-paper.pdf, 2013.

[7]. D. MacVittie, "Intro to load balancing for developers — The algorithms‖, https://devcentral.f5.com/blogs/us/introto-load-balancing-for-developers-ndash-the-algorithms, 2012.

[8]. DoddiniProbhuling L. ―Load Balancing Algorithms In Cloud Computing‖, International Journal of Advanced Computer and Mathematical Sciences, ISSN 2230-9624. Vol. 4, Issue 3, pp. 229-233, 2013.

[9]. Naimesh D. Naik and Ashilkumar R. Patel ―Load Balancing Under Bursty Environment for Cloud Computing‖, International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181, Vol. 2, Issue 6, pp. 17 – 26, June – 2013.

[10]. Kaviani, Nima, Eric Wohlstadter, and Rodger Lea. "MANTICORE: A framework for partitioning software services for hybrid cloud." In Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on, pp. 333-340, 2012.