S. Vigneshwari *et al*, International Journal of Computer Science and Mobile Applications,
Vol.1 Issue. 3, September- 2013, pg. 45-49 **ISSN: 2321-8363**

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND MOBILE APPLICATIONS - IJCSMA

BALANCING CHUNKS FOR DISTRIBUTED FILE SYSTEMS IN CLOUDS BY USING LOAD REBALANCING ALGORITHM

S. Vigneshwari[1], B. Sunitha Devi[2]
[1]*M.Tech Student, CSE Dept, CMR Institute of Technology, Hyderabad, A.P*
Email-id: vigneshwarireddy@yahoo.com
[2]*Associate Prof., CSE Dept., CMR Institute of Technology, Hyderabad, A.P*
Email-id: sunithabigul@gmail.com

Abstract
Currently most of the cloud applications method great deal of knowledge to supply the required results. Information volumes to be processed by cloud applications are growing a lot of quicker than computing power. These growth difficulties on new approaches for process and analyzing the knowledge. The project explores the employment of Hadoop Map Reduce framework to execute scientific workflows within the cloud. Cloud computing provides monumental clusters for economical giant division and information analysis. In such file systems, a file is divided into variety of file chunks allotted in distinct nodes so Map Reduce tasks will perform in parallel over the nodes to form resource utilization effective and to enhance the interval of the task. In giant failure prone cloud environments files and nodes are dynamically created, replaced and extra within the system owing to that a number of the nodes are over loaded whereas some others are underneath loaded. It results in load imbalance in distributed filing system. to beat this load imbalance drawback, a completely distributed Load rebalancing algorithmic program has been applied, that is active in nature doesn't contemplate the previous state or behaviour of the system and it solely depends on this behaviour of the system and approximation of load, comparison of load, stability of various system, performance of system, interaction n between the nodes, nature of load to be transferred, choice of nodes and network traffic. The present Hadoop implementation assumes that computing nodes in an exceedingly cluster are homogenized in nature. The performance of Hadoop in heterogeneous clusters wherever the nodes have completely different computing capability is additionally tested.

Keywords: Cloud Computing; distributed Hash tables; load rebalancing

1. Introduction

Cloud computing could be a comparatively new approach of pertaining to the employment of shared computing resources, associated it's a completely different to having native servers handle applications. Cloud computing teams along giant numbers of pc servers associated alternative resources and usually offer their combined capability on an on-demand, pay-per-cycle basis while not refined readying and management of resources. The tip users of a cloud computing network sometimes don't have any plan wherever the servers are physically situated; they solely spin up their application and begin operating. In an exceedingly giant cloud we are able to add thousands of nodes along. The most aim is to allot files to those nodes while not creating significant load to any of the nodes, for that files are divided into completely different modules. Another objective is to scale back the network inconsistencies and network traffic as a result of the unbalancing of masses. The reduction of network inconsistency can result in maximization of network information measure so

{so many|numerous|such a big amount of|such a giant amount of|such a lot of} large applications will run in it. Owing to quantifiability property we are able to add, delete, and update new nodes so it supports non-uniformity of the system. To enhance the aptitude of nodes we tend to use Distributed filing system in Cloud Computing Applications. In such file systems the most functionalities of nodes is to serve computing and storage functions. If we wish to store a file into the system foremost we'll divide the file into completely different modules and store it in several nodes. Therefore we tend to introduced new load rebalancing algorithmic program to avoid of these disadvantages. Once analyzing the prevailing system clouds place confidence in central nodes to balance the masses of storage nodes, there comes the performance bottleneck as a result of the failure of central nodes results in the failure of whole system and it'll results in several technical and purposeful difficulties.

2. Methods

There are four completely different modules during this system. They're shown below.

- Chunk Creation
- DHT Formulation
- Load Rebalancing algorithmic program
- Duplicate Management

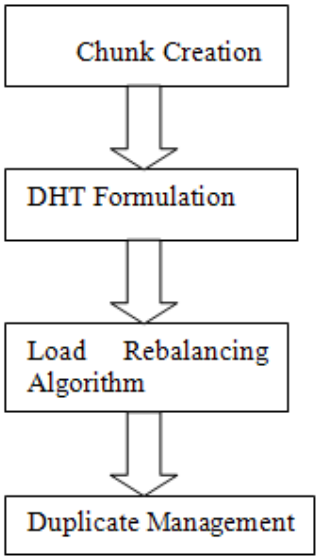


Fig 1. Load Rebalancing Process

2.1. Chunk Creation

File is split into variety of chunks allotted in distinct nodes so Map scale back Tasks may be performed in parallel over the nodes. The load of a node is usually proportional to the amount of file chunks the node possesses. as a result of the files in an exceedingly cloud may be willy-nilly created, deleted, and appended, and nodes may be upgraded, replaced and extra within the filing system, the file chunks aren't distributed as uniformly as potential among the nodes. The most objective is to allot the chunks of files as uniformly as potential among the nodes such no node manages associate unnecessary variety of chunks.

2.2. DHT Formulation

The storage nodes are structured as a network supported distributed hash tables (DHTs), e.g., discovering a file chunk will merely ask speedy key search in DHTs, providing a novel handle is assigned to every file chunk. DHTs modify nodes to self-organize and Repair whereas perpetually giving search practicality in node dynamism, simplifying the system provision and management. The chunk servers in our proposal are organized as a DHT network. Typical DHTs guarantee that if a node leaves, then its regionally hosted chunks are faithfully migrated to its successor; if a node joins, then it allocates the chunks whose IDs at once precede the connection node from its successor to manage.

2.3. Load Rebalancing Formula

In load rebalancing algorithmic program, every chunk server node initial estimate whether or not it's underneath loaded (light) or overladen (heavy) while not world information. A node is lightweight if the amount of chunks it hosts is smaller than the brink. Load positions of a sample of arbitrarily elite nodes.

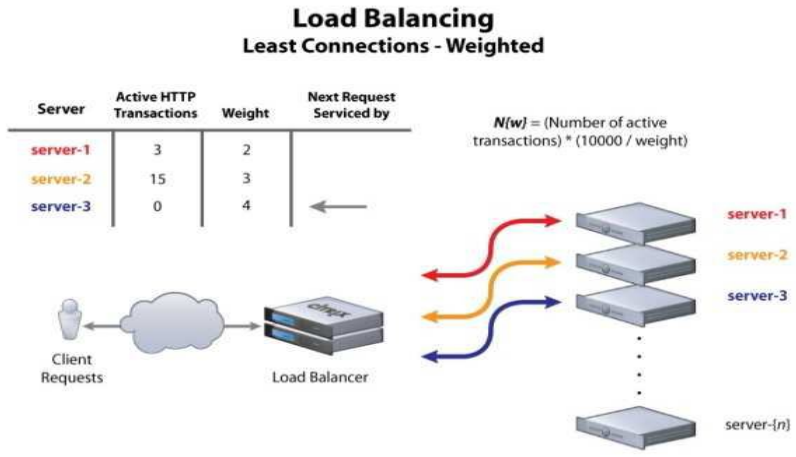


Fig 2: the overall variety of messages generated by a load rebalancing algorithmic program

Specifically, every node contacts variety of arbitrarily elite nodes within the system and builds a vector denoted by V. A vector consists of entries, and every entry contains the ID, network address and cargo standing of a arbitrarily elite node. Fig. two shows the overall variety of messages generated by a load rebalancing algorithmic program.

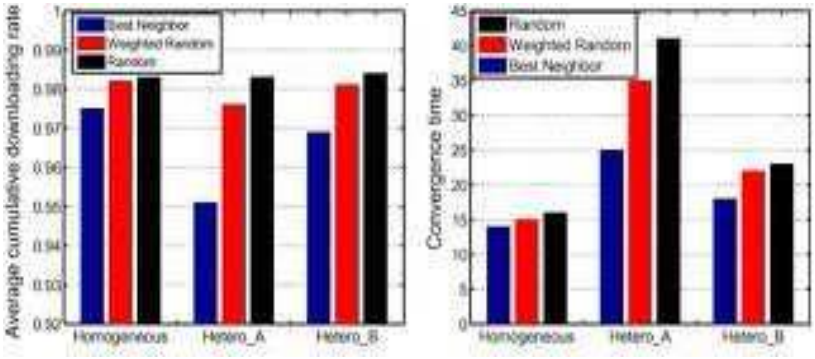
2.4. Duplicate Management

Google GFS and Hadoop HDFS , a persistent form of replicas for each file module area unit maintained in distinct nodes to reinforce file proximity with relevancy node failures and departures. Our current load equalisation rule does not treat replicas clearly. It's unlikely that two or further replicas area unit placed in an exceedingly consistent node because of the random nature of our load rebalancing rule. Additional specifically, each less loaded node samples form of nodes, each elect with a chance of 1/n, to share their tons.

3. Distributed classification system

There are many demonstrably economical load equalization for distributed file's protocols for distributed information storage in P2P systems. Additional details and analysis may be found in an exceedingly theory. Our algorithms are straightforward, and simple to implement in distributed files therefore an evident next analysis step ought to be a sensible analysis of those schemes.

First, it would be potential to additional improve the consistent hashing theme as mentioned at the tip of our vary search arrangement. Distributed doesn't simply generalize to quite one order. For instance (Fig.2) once storing music files, one would possibly need to index them by each creative person and song title, permitting lookups in keeping with 2 orderings. Since our protocol rearranges the things in keeping with the ordering, doing this for 2 orderings at a similar time looks tough. A simple, however unsophisticated, answer is to set up not the things themselves, however simply store tips to them on the nodes. This needs way less storage, and Makes it potential to keep up 2 or additional orderings directly. Lastly, permitting nodes to settle on random addresses in our item equalization protocol for distributed file's makes it easier for cruel Nodes to disturb the operation of the P2P network.



Network Setting

Fig 3: The common downloading rate and Convergence time

4. Conclusion

A unique load equalization algorithmic program to alter the load rebalancing drawback in large-scale, dynamic, and distributed file systems in clouds has been given. Our proposal tries to balance the masses of nodes and scale back the demanded movement price the maximum amount as potential, whereas taking advantage of physical network neighbourhood and node non-uniformity. Within the absence of representative real workloads i.e., the distributions of file chunks in an exceedingly large-scale storage system within the property right, we've got investigated the performance of our proposal and compared it against competitor algorithms through synthesized probabilistic distributions of file chunks. The synthesis workloads assay the load equalization algorithms by making a couple of storage nodes that are heavily loaded.

5. Future Work

In complicated and enormous systems, there's a unimaginable want for load equalization. For simplifying load equalization globally e.g. in a cloud, one issue which might be done is, using techniques would act at the elements of the clouds in such the simplest way that the load of the total cloud is balanced. Cloud Computing could be a huge idea and cargo equalization plays a awfully vital role just in case of Clouds. there's a large scope of improvement during this space. The performance of the given algorithms may be inflated by varied completely different parameters.

References

[1] J. Dean and S. Ghemawat, "MapReduce: Simplified processing on giant Clusters," in Proc. 6th Symp. software system style and Implementation (OSDI'04), Dec. 2004, pp. 137–150.
[2] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The Google filing system," in Proc. nineteenth ACM Symp. in operation Systems Principles (SOSP'03), Oct. 2003, pp. 29–43.
[3] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: a scalable Peer-to-Peer search Protocol for web Applications," IEEE/ACM Trans. Netw., vol. 11, no. 1, pp. 17–21, Feb. 2003.
[4] J. W. Byers, J. Considine, and M. Mitzenmacher, "Simple Load equalization for Distributed Hash Tables," in Proc. first Int'l Workshop Peer-to-Peer Systems (IPTPS'03), Feb. 2003, pp. 80–87.
[5] Y. Zhu and Y. Hu, "Efficient, Proximity-Aware Load equalization for DHTBased P2P Systems," IEEE Trans. Parallel Distrib. Syst., vol. 16, no. 4, pp. 349–361, Apr. 2005.
[6] A. Rao, K. Lakshminarayanan, S. Surana, R. Karp, and I. Stoica, "Load equalization in Structured P2P Systems," in Proc. ordinal Int'l Workshop Peerto- Peer Systems (IPTPS'02), Feb. 2003, pp. 68–79.
[7] D. Karger and M. Ruhl, "Simple economical Load equalization Algorithms for Peer-to-Peer Systems," in Proc. sixteenth ACM Symp. Parallel Algorithms and Architectures (SPAA'04), June 2004, pp. 36–43.

Authors' Biography

S. Vigneshwari had done B.Tech from Narsimha Reddy Engineering College, maisammaguda. She is M.Tech Student in CSE Department of CMR Institute of Technology, Hyderabad. She is currently working for her M.Tech. Research Project work under the guidance of Mrs.B.Sunitha Devi. Her areas of interest include Cloud computing, Computer Networks, and Programming Languages.

B. Sunitha Devi had done B.Tech from J.N.T.U, Hyderabad, and she had done M.Tech from J.N.T.H.C.E.H University, Hyderabad. She is currently working as Associate Professor in CSE Department of CMR Institute of Technology, Hyderabad. Her areas of interest are Network Security, Data Mining, Image Processing, and Neural Networks for various fields

