ISSN: 2321-7782 (Online)
Volume 2, Issue 3, March 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Paper / Case Study
Available online at: www.ijarcsms.com

Load Balancing On Cloud Data Centers Using Adaptive Overlapped Chained Declustering

Snehal a. Tambat[1]
Dept of Computer Engg
Dr. D.Y.Patil IET
Ambi-Pune – India

Rachana R. Koshti[2]
Dept of Computer Engg
Dr. D.Y.Patil IET
Ambi-Pune – India

Nilima B. Bhosale[3]
Dept of Computer Engg
Dr. D.Y.Patil IET
Ambi-Pune – India

Arati A. Khamkar[4]
Dept of Computer Engg
Dr. D.Y.Patil IET
Ambi-Pune – India

Abstract: There is a big growth in cloud computing as compared to last few years. In future we would have only our desktop with us with all our data on cloud. We would directly interact with the cloud. With the increase in the users of cloud resources comes the concept of load balancing on cloud data centers. Load balancing is a key challenge for the cloud service providers in order to avoid the data lose of users, any type of failures, and other security issues.

There are many types of load balancing aspects, in this paper we are more concern to balancing load on virtual machines.

Keywords: Cloud Computing, Load Balancing, Adaptive Overlapped Chained Declustering, Hadoop(MapReduce).

I. INTRODUCTION

Cloud computing is a start of IT towards new future. Cloud Computing comes into picture when we think of the IT sector and its expanding needs for memory storage and processing speed. Cloud service providers provide the users collection of computing services and resources clubbed together in a single infrastructure on pay-per-use basis.

So to deal with this load of users of data access, service providers distribute the load among number of virtual machines. One of the advantages of distributing the data over virtual machines is reduction in energy consumption, loss of user data and avoiding deadlocks.

In this paper the technique used for distributing data over virtual nodes is Adaptive Overlapped Chained Declustering. This strategy provides for higher availability of data and enhances performance in case of failure of virtual nodes. This technique does not require any additional hardware for its implementation.

Chained Declustering can be implemented using MapReduce concept of Hadoop. MapReduce is a programming model for processing large data sets with a parallel, distributed algorithm on a cluster.

II. SYSTEM OBJECTIVES

The intension of this project is:

In today's world many organizations prefer to use cloud for storage of their data due to its simple and flexible architecture. There is exponential growth of cloud users and rapid expansion of data centers. Here arises the main challenge of load balancing.

Load balancing is essential concept required to distribute dynamic workload across multiple virtual nodes to ensure no single node gets overwhelmed. The advantage of load balancing is optimal utilization of resources which in turn enhances the system performance and leads to maximum user satisfaction.

III. LITERATURE SURVEY

Existing System

In this section we describe the techniques available in load balancing.

1. Tandem's Mirrored Disks Architecture :

The hardware structure of a Tandem system [Borr81] consists of one or more clusters that are linked together by a token ring. Each cluster contains 2 to 16 processors with multiple disk drives. The processors within a cluster are interconnected with a dual high speed (˜20 Mbyte/sec) bus. Each processor has its own power supply, main memory, and I/O channel. Each disk drive is connected to two I/O controllers, and each I/O controller is connected to two processors, providing two completely independent paths to each disk drive. Furthermore, each disk drive is "mirrored" (duplicated) to further insure data availability.

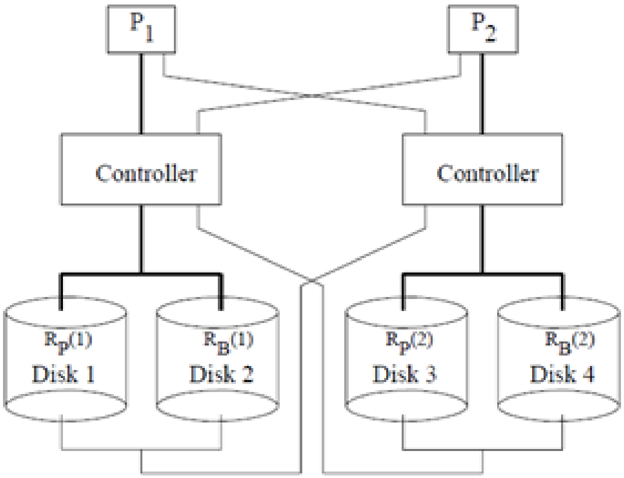

Fig. 1 Data Placement with Tandem's Mirrored Disk Scheme.

Limitations:

While the use of mirrored disk drives offers the highest level of availability, it does a poor job of distributing the load of a failed processor.

2. Teradata's Data Clustering Scheme

In this case, one copy is designated as the primary copy and the other as the backup or fallback copy. Fragments of the primary and backup copies are termed primary and backup fragments respectively. Each primary fragment is stored on one node. For backup fragments, Teradata employs a special data placement scheme. If the cluster size is N, each backup fragment will be subdivided into N-1 subfragments. Each of these subfragments will be stored on a different disk within the same cluster other than the disk containing the primary fragment.

Limtations:

Teradata's scheme provides a tradeoff between availability and performance in the event of a failure. As the cluster size is increased, the probability of two failures rendering data unavailable increases while the imbalance in workloads among the processors in the event of a failure decreases.

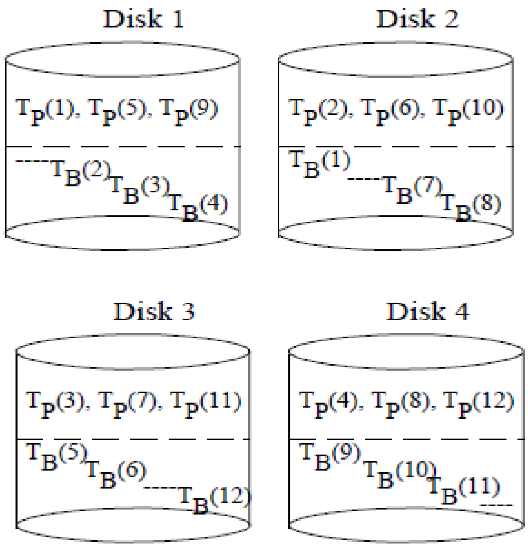

Fig. 2 Teradata's Data Clustering Scheme.

3. RAID's Data Storage Scheme

In the RAID data storage scheme [Patt88, Gibs89], an array of small, inexpensive disks is used as a single storage unit termed a group. Instead of replicating data on different drives, this strategy stores check (parity) bytes for recovering from both random single byte failures and single disk failures. This scheme interleaves (stripes) each data block, in units of one disk sector, across multiple disk drives such that all the sectors in a single block can be accessed in parallel, resulting in a significant increase in the data transfer rate.

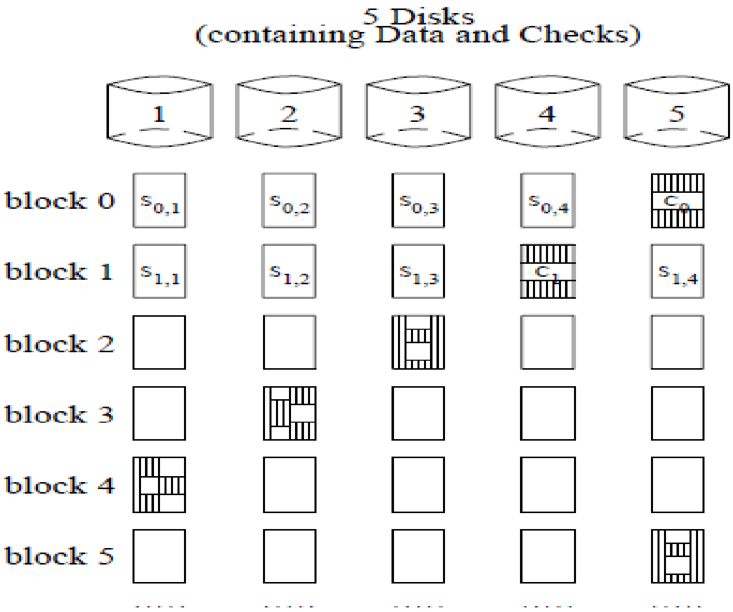

Fig. 3 Data placement in RAID

Limitations:

RAID emphasizes the importance of disk space over performance.

IV. DESIGN AND IMPLEMENTATION

What is to be implemented?

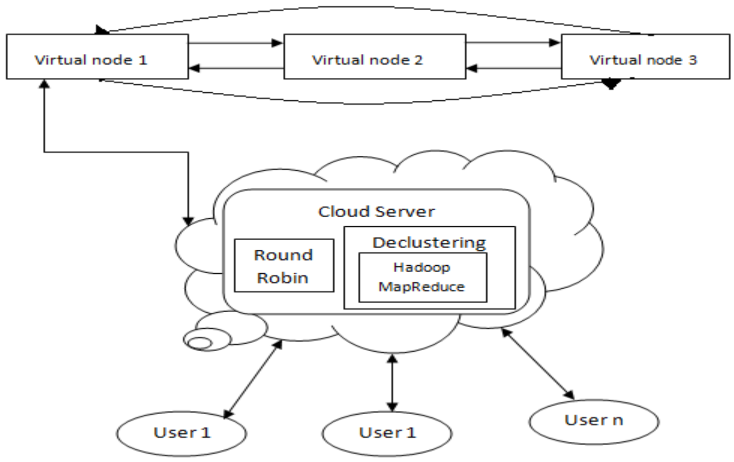

Fig. 4 Proposed System

Implementing Load Balancing –

1. **Declustering strategy**

MapReduce

MapReduce is a programming model for processing large data in parallel distributed algorithm on cluster. MapReduce is a mixture of two procedures, namely Map() and Reduce().

MapReduce libraries have been written in many programming languages with different levels of optimization. Apache Hadoop is a popular open source for MapReduce implementation.

Map (): The data when uploaded on server node. The server node equally distributes data on virtual node. This distribution is done on the basis of number of virtual nodes, (i.e. Size of the file / Number of virtual nodes). Due to this no single node gets overloaded.

Reduce (): When user wish to download the file, the server perform Reduce() operation. In this the server collects the data from all the virtual machines, merges the parts and serves the user with the original uploaded file.

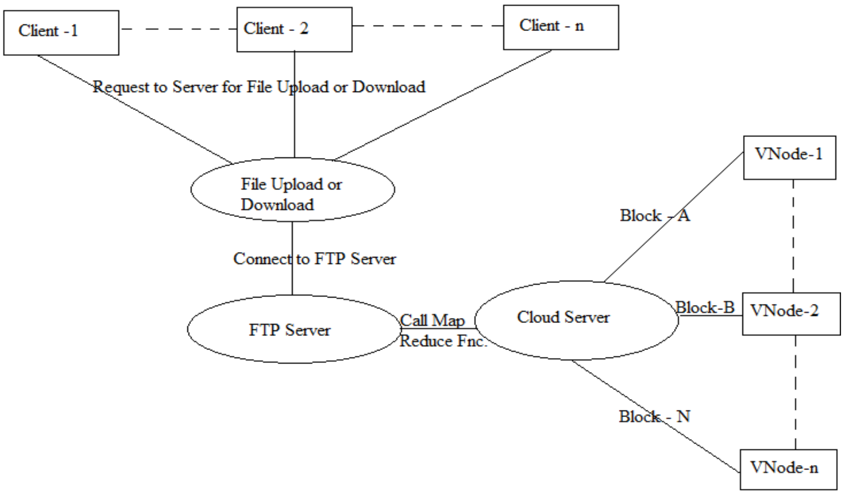

Fig. 5 Declustering strategy using MapReduce

Comparison between Existing System and Proposed System

In the existing system the server allocates the load on the virtual machines on the basis of some priority, or the nearest hop technique. Every virtual node maintains a counter number; the counter number indicates the request load on the node. The node with the lowest number, the server allocates the client data on that respective node. Another technique is by using nearest hop technique. The server finds the virtual node that is near to the user to store the user's data.

The drawback of the existing system is, a single particular node may get loaded. If the virtual node gets failed there is no provision of backup of user data.

In the proposed system the limitations of the existing system are overcome. Since the data is divided equally no single node gets overloaded. In case of failure the backup of the data can be collected from the adjacent virtual nodes. This helps in improving the efficiency and performance of the cloud.

V. Algorithm

1. Adaptive Overlapped Chained Declustering

The file that is split using MapReduce is distributed over number of virtual machines and the backup of each split on a particular virtual machine is saved on its adjacent virtual machine hence, called Adaptive Overlapped Chained Declustering.

When the server finds a particular virtual machine is failed, the backup of the split is collected from its adjacent virtual node.

2. Round Robin Algorithm

After the file is split the server stores the data on the virtual machines. In case of failure of any virtual machine the split data is stored on the next virtual node.

VI. Conclusion

The proposed system is able to manage the data on the cloud. The backup facility increases the degree of data availability in the system. Data of one virtual machine is stored on its adjacent virtual machine, if any of virtual machine gets failed then also intended data can be easily recovered due to backup facility. Due to use of MapReduce facility of Hadoop, it enables scaling of applications across large clusters of machines comprising thousands of nodes, with fault-tolerance built-in for ultra-fast performance.

References

1. International Journal of Advanced Research in Computer Science and Software Engineering Research Paper "Load Balancing On Cloud Data Centres" Dr. Hemant S. Mahalle Prof. Parag R. Kaveri Dr.Vinay Chavan.
2. "Chained Declustering: A New Availability Strategy for Multiprocssor Database machines" Hui-I Hsiao, David J. DeWitt Computer Sciences Department University of Wisconsin.
3. "Enhanced Load Balancing Approach to Avoid Deadlocks in Cloud" Rashmi. K. S, Suma. V, Vaidehi. M Research and Industry Incubation Center, Dayananda Sagar Institutions, Bangalore, India.
4. "Enhanced Equally Distributed Load Balancing Algorithm for Cloud Computing" Shreyas Mulay, Sanjay Jain
5. International Journal of Computers & Technology " Analysis of Load Balancing Techniques in Cloud Computing" Amandeep Kaur Sidhu1, Supriya Kinger2.
6. "A Load Balancing Model Based on Cloud Partitioning for the Public Cloud" Gaochao Xu, Junjie Pang, and Xiaodong Fu.
7. Journal of Information Systems and Communication ISSN: 0976-8742 & E-ISSN: 0976-8750, Volume 3, Issue 1, 2012 "Existing Load Balancing Techniques In Cloud Computing: A Systematic Review" Nidhi Jain Kansal* And Inderveer Chana.
8. 2011 International Conference on Computer and Software Modeling IPCSIT vol.14 (2011) IACSIT Press, Singapore "Availability and Load Balancing in Cloud Computing" Zenon Chaczko 1, Venkatesh Mahadevan 2, Shahrzad Aslanzadeh 1 and Christopher Mcdermid1.
9. ARPN Journal of Systems and Software ©2009-2012 AJSS Journal. All rights reserved "Cloud Computing Vs. Grid Computing" 1Seyyed Mohsen Hashemi, 2Amid Khatibi Bardsiri.
10. International Journal of Engineering Research & Technology (IJERT)Vol. 2 Issue 2, February- 2013 ISSN: 2278-0181 " A Novel Survey on Load Balancing in Cloud Computing" Aarti Khetan Vivek Bhushan Subhash Chand Gupta.
11. International Journal of Engineering Research & Technology (IJERT)Vol. 1 Issue 10, December- 2012 ISSN: 2278-0181 "Load Balancing The Essential Factor In Cloud Computing" Mr. Jayant Adhikari, Prof. Sulabha Patil.