**Available Online at www.ijcsmc.com**

**International Journal of Computer Science and Mobile Computing**

**A Monthly Journal of Computer Science and Information Technology**

**ISSN 2320–088X**
**IMPACT FACTOR: 5.258**

***IJCSMC, Vol. 5, Issue. 3, March 2016, pg.486 – 491***

# Data Processing in Cloud with AVL Structure and Bootstrap Access Control

**K.Dhivya Bharathi, R.Geetha, G.Sathya, Dr. R.Kalpana**
IFET College of Engineering, Villupuram 605602, Tamil Nadu, India
geethaifet@gmail.com

***Abstract- Corporate networks store their data in the shared knowledge plane which is cloudy. But they have to decide which part of the data would be visible for which users. To accomplish this task, the bootstrap access control mechanism is used which decides the accessibility of the user while entering the database. Also, if the data are processed and stored in perfect data structure, then data mining and data processing would be easier. So, the data are processed with map reduce algorithm and stored in a Balanced Overlay Network (BATON). BATON uses AVL tree structure for storing data. Also, it supports peer to peer system in which a separate server is provided for each company in the corporate network and in each server, the required redundancy is maintained. "Pay as you go" pricing policy is followed in this system which is a two way (Service provider and corporate network) beneficial pricing policy. In this policy, the user will be charged by categorizing his needs based on a range of memory required, duration and security policy.***

## 1. INTRODUCTION

The corporate networks such as supply chain networks have to share relevant information among the companies which are in collaboration in the same industry sector. So, they have to store, retrieve and process a huge amount of data. This requires huge databases and servers. Hence they choose third party data warehouse to store their data. But this has several threats like information disclosure, quality of service and data leaks. Also a data warehouse is static that is its storage procedure is constant. With the advent of cloud computing. so we are moving to the AVL structure for storage process and map reduce algorithm for efficient retrieval of the well-structured stored data, which will minimizes the retrieving time of the data.

## 2. RELATED WORKS

This paper HADOOPDB: AN ARCHITECTURAL HYBRID OF MAPREDUCE AND DBMS TECHNOLOGIES FOR ANALYTICAL WORKLOADS says about The production environment for analytical data management

applications is rapidly changing. Many enterprises are shifting away from deploying their analytical databases on high-end proprietary machines, and moving towards cheaper, lower-end, commodity hardware, typically arranged in a shared-nothing MPP architecture, often in a virtualized environment inside public or private“ clouds”. At the same time, the amount of data that needs to be analyzed is exploding, requiring hundreds to thousands of machines to work in parallel to perform the analysis.

This paper A P2P TECHNOLOGY BASED ON RELATIONAL DATABASE IN CLOUD COMPUTING says about A corporate network is a group of computers, connected together in a building or in a particular area, which are all owned by the same company or institutions. It provides the flexible, economical & Scalable platform. Data management system including scalability, flexibility, security, Data Sharing, better performance, no of peers easily added & remove it's challenging for corporate network. Centralized data processing where data processing supported by one clusters of computers means all data stored on centralized platform. In centralized data processing it's necessary to handle the computers overload problem; this problem is overcome in using Best Peer++ concept. Best peer ++ achieves linear scalability throughput with respect to the number of peer nodes, this problem is overcome using bootstrap and adaptive query processing algorithm.

This paper BENCHMARKING CLOUD SERVING SYSTEMS WITH YCSB says about While the use of Map Reduce systems (such as Hadoop) for large scale data analysis has been widely recognized and studied[7], we have recently seen an explosion in the number of systems developed for cloud data serving. These newer systems address “cloud OLTP” applications, though they typically do not support ACID transactions. Examples of systems proposed for cloud serving use include Big Table, PNUTS, Cassandra, HBase, Azure, CouchDB, SimpleDB, Voldemort, and many others. Further, they are being applied to a diverse range of applications that differ considerably from traditional (e.g., TPC-C like) serving workloads. The number of emerging cloud serving systems and the wide range of proposed applications, coupled with a lack of apples to apples performance comparisons, makes it difficult to understand the tradeoffs between systems and the workloads for which they are suited. We present the Yahoo! Cloud Serving Benchmark (YCSB) framework, with the goal of facilitating performance comparisons of the new generation of cloud data serving systems. We define a core set of benchmarks and report results for four widely used systems: Cassandra, HBase Yahoo!’s PNUTS, and a simple shared MySQL implementation. We also hope to foster the development of additional cloud benchmark suites that represent other classes of applications by making our benchmark tool available via open source. In this regard, a key feature of the YCSB framework/tool is that it is extensible it supports easy definition of new workloads, in addition to making it easy to benchmark new systems.

## 3. MAP REDUCE ALGORITHM

**Map Reduce** is a programming model and an associated implementation for processing and generating large data sets with a parallel, distributed algorithm on a cluster[1],[2].A Map Reduce program is composed of a **Map()** procedure that performs filtering and sorting (such as sorting students by first name into queues, one queue for each name) and a **Reduce()** procedure that performs a summary operation[6] (such as counting the number of students in each queue, yielding name frequencies). The "Map Reduce System[5]"(also called "infrastructure" or "framework") orchestrates the processing by marshalling the distributed servers, running the various tasks in

parallel, managing all communications and data transfers between the various parts of the system, and providing for redundancy and fault tolerance[8]. The model is inspired by the map and reduce functions commonly used in functional programming, although their purpose in the Map Reduce framework is not the same as in their original forms[7]. The key contributions of the Map Reduce framework are not the actual map and reduce functions, but the scalability and fault-tolerance achieved for a variety of applications by optimizing the execution engine once. As such, a single-threaded implementation of Map Reduce (such as MongoDB) will usually not be faster than a traditional (non-Map Reduce) implementation; any gains are usually only seen with multi-threaded implementations. Only when the optimized distributed shuffle operation (which reduces network communication cost) and fault tolerance features of the Map Reduce framework come into play, is the use of this model beneficial. Map Reduce libraries have been written in many programming languages, with different levels of optimization. A popular open-source implementation is Apache Hadoop[2]. The name Map Reduce originally referred to the proprietary Google technology, but has since been generalized.

## 4. AVL TREE STRUCTURE

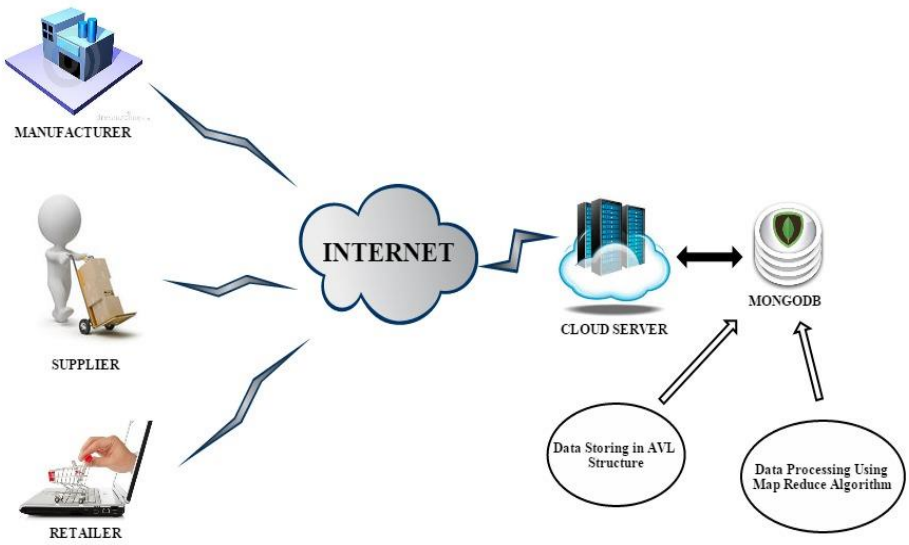
AVL trees are height balanced binary searched trees. Balance factor of a node can be calculated as height(left subtree) -height(right subtree).an AVL tree has a balance factor calculated at every node for every node heights of left and right subtree can differ by no more than 1.store current heights in each node .you can either keep a height or just the difference in height, i.e. the balance factor; this has to be modified on the path of insertion even if you don't perform rotations .once you have perform the rotation(single or double)you won't need to go back up the tree**.** Insert operation may cause balance factor to become 2 or –2 for some node only nodes on the path from insertion point to root node have possibly changed in height So after the Insert, go back up to the root node by node, updating heights If a new balance factor is 2 or –2, adjust tree by *rotation* around the node.

## 5. PROBLEMS IN EXISTING SYSTEM

The corporate networks stores their bulk data in a third party data warehouse which may have security threats and high cost. Also access control mechanism is not well defined for various categories of users like suppliers, manufacturers and retailers. The data are stored in an unstructured manner so that retrieval of data and processing them are tedious. There is no processing like sorting or clustering before storing the data from the server. So, while retrieving, the complexity of the system will be increased because it has to search for the whole database. The scalability of the system is very low since it cannot scale up to thousands of participants. The storage of data in the data warehouse system entails non trivial costs, including hardware/software investment and high maintenance cost. The inside processing of data marts and classification of fact tables and dimension tables is complex tasks when we store data in the data warehouse. The system is not supported for heterogeneous environment that is participants of the network using different platforms cannot be supported.

## 6. SYSTEM ARCHITECTURE

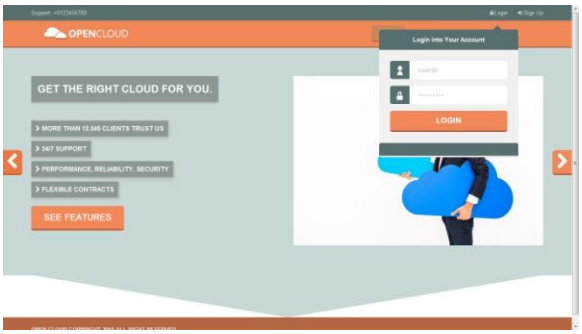



## 7. ADVANTAGES OF PROPOSED SYSTEM

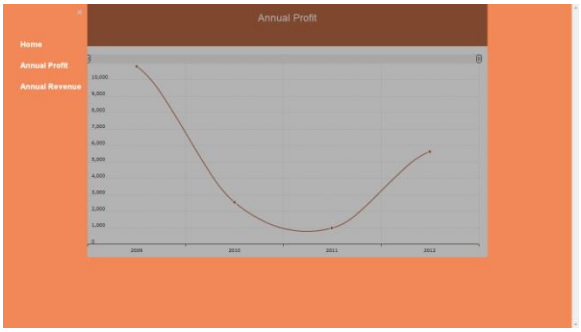
The data to be shared in the in the corporate network are stored in the cloud database after proper processing. The data are processed using map reduce algorithm before storing in the cloud database. The data are structured and stored in the balanced tree overlay network, which uses the AVL tree structure. The retrieval of data from these databases is simple since they are stored in a well structured manner. The access control for various categories of participants of the corporate network can be implemented using bootstrap server. The scalability of the system is high as it can hold thousands of participants without any complexity.

## 8. EXPERIMENTAL RESULTS








## 9. CONCLUSION

Thus in this paper we conclude that, the data will be easily stored and retrieved efficiently from the cloud. data can be stored in the mongoDB by using the AVL structure and retrieved by using the map reduce technique. The both AVL tree structure and the map reduce technique is used in this paper. bootstrap access control is used for the privacy of the organization and this will be enabled once the customer enters in to the homepage. It shows the information according to their usage, the boot strap access will decide which part of the information should be visible for which user, according to their category. Huge number of data can be stored and retrieved in an efficient manner since we are storing in a well structured manner.

REFERENCES

[1] A Workload Model for MapReduce by Thomas A. de Ruiter, Master‟s Thesis in Computer Science, Parallel and Distributed Systems Group Faculty of Electrical Engineering, Mathematics, and Computer Science. Delft University of Technology, 2nd June 2012.
[2] HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads by Azza Abouzeid1, Kamil BajdaPawlikowski1,Daniel Abadi1, Avi Silberschatz1, Alexander Rasin2 1Yale University, 2Brown University, Sep 2013
[3] Benchmarking Cloud Serving Systems with YCSB Brian F. Cooper, Adam Silberstein, Erwin Tam, Raghu Ramakrishnan, Russell Sears Yahoo! Research Santa Clara, CA, USA, may 2013
[4] A P2P TECHNOLOGY BASED ON RELATIONAL DATABASE IN CLOUD COMPUTING by Akshata Desai, Zeenat Kalambkar, Uma Madake, Megha Jadhav. June 2012
[5] Hive A Warehousing Solution Over a MapReduce Framework
Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Suresh Anthony, Hao Liu, Pete Wyckoff and Raghotham Murthy, july 2013
[6]J. Dean and S.Ghemawat,(2004) "MapReduce: simplified data processing on large clusters", Google Inc. In OSDI'04: Proceeding of the 6th conference on Symposium on Operating Systems Design & Implementation, San Francisco, CA.
[7] S.N.Srirama, P.Jakovits, E.Vainikko, (2011) "Adapting scientific computing problems to clouds using MapReduce", Future Generation Computer Systems, Vol. 28, No. 1, pp184-192.
[8] R.Lammel, Data Programmability Team, Microsoft Corp, Redmond, (2007), "Google's MapReduce programming model Revisited", WA, USA, Available online at http://www.sciencedirect.com