EFFICIENT DATA STORAGE AND RETRIEVAL IN BIGDATA USING VIRTUAL SHUFFLING

[1] P.Annadurai

[1]Assistant Professor
[1]Computer Science and Engineering,
[1]Sri Sai Ram Institute of Technology, Chennai, India

Abstract—Big data is a storage infrastructure that is designed specifically to store, manage and retrieve massive amount of data. Big Data is a large or complex data sets in which traditional data processing applications are inadequate. Big Data need to scale very rapidly and elastically, whenever and wherever we want. Instead of concentrating on the entire data sets, only the required portion of data sets can be handled efficiently. To overcome the scalability problem and to efficiently handle fault tolerance mechanism in Big Data, shuffling strategy can be deployed. It is achieved through a combination of three techniques including three level section table (i.e.) section general directory, section intermediate directory and section table directory. Server will act as the intermediate layer that will process the user's request and get concerned data from database. ABE is regarded as one of the most important technology for data access control. Hence, shuffling strategy significantly speeds up data movement in MapReduce and achieves faster job execution. Time taken for data retrieval from respective directories is also reduced.

Keywords-Big data, Virtual shuffling, Data partitioning, Attribute based encryption.

I. INTRODUCTION

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. A Map Reduce program is composed of a **Map()** procedure (method) that performs filtering and sorting (such as sorting students by first name into queues, one queue for each name) and a **Reduce()** method that performs a summary operation (such as counting the number of students in each queue, yielding name frequencies)[9]. The Map Reduce System (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[1].

Big data able to scale very rapidly and elastically, wherever we want. HDFS cluster has 2 type of nodes: NameNode(master), DataNode(workers). NameNode holds files system meta data in the memory. When memory on the NameNode is full there is no further chance of increasing the cluster capacity.

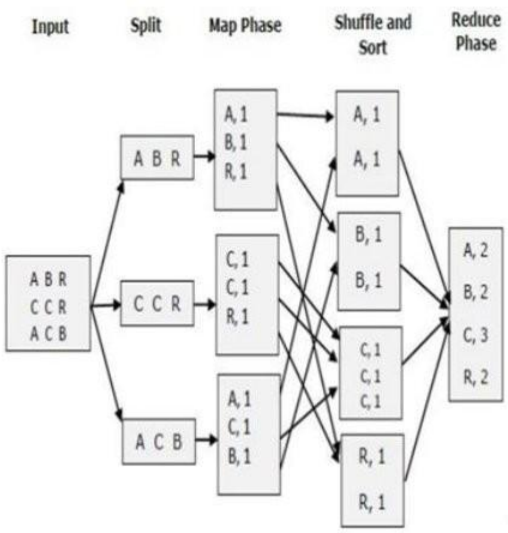


Figure1- Map &Reduce Example

Information of a person with large data sets need to be secret, the person might not want any other person to know about them. In BigData, whenever failure occurs the damage done should be within acceptable threshold rather than beginning the whole task. Hence we propose shuffling strategy to overcome scalability problem and to efficiently handle fault tolerance mechanism. Here, instead of moving entire data into memory of Name Node, only the required data is moved into local disk. For, Privacy and security concerns ABE access control can be deployed which will provide privileged access with the data. To implement fault tolerance handling mechanism in BigData, JobTracker and TaskTracker is used. If a TaskTracker fails to communicate with JobTracker for a period of time, JobTracker will assume that TaskTracker has crashed. JobTracker will choose another TaskTracker to re-execute all map tasks, that were previously executed at the failed TaskTracker.

II. LITERATURE SURVEY

Hadoop is a infrastructure which is currently being used for big data management. The foundation of Hadoop consists of Hadoop Distributed File System (HDFS). HDFS presents a client-server architecture comprised of a NameNode and many DataNodes. The NameNode stores the metadata for the DataNodes and DataNode stores application data. DataNode performs creation, deletion and copy of block under the NameNode's command. They store and retrieve blocks when they are told to (by clients or the NameNode), and they report back to the NameNode periodically with lists of blocks that they are storing. The NameNode holds file system metadata in memory, and thus the limit to the number of files in a file system is governed by the amount of memory on the NameNode. Thus when the memory on NameNode is full there is no further chance of increasing the cluster capacity[2].The NameNode keeps a reference to every file and block in the file system in memory, which means that on very large clusters with many files, memory becomes the limiting factor for scaling. Thus when a NameNode memory is full it becomes difficult to just add another node in the cluster for further storage needs as the NameNode will not be able to handle this extra node and file system metadata that will be generated from it. This raises the NameNode scalability issue.

Fault tolerance

Failures happen for large scale distributed systems such as Hadoop clusters. Hadoop provides basic support for failure tolerance. With the incoming of new technologies like cloud computing and BigData it is always intended that whenever the failure occurs the damage done should be within acceptable threshold rather than beginning the whole task from the scratch. Fault-tolerant computing is extremely hard, involving intricate algorithms. It is simply not possible to devise absolutely foolproof, 100% reliable fault tolerant machines or software. Thus the main task is to reduce the probability of failure to an "acceptable" level. Unfortunately, the more we strive to reduce this probability, the higher the cost. There could be some tasks which might be recursive in nature and the output of the previous computation task is the input to the next computation. Thus restarting the whole computation becomes cumbersome process.

Privacy and Security

A security and privacy concern in big data becomes more and more accessible. Attempts to anonymous specific data are not successful in protecting privacy because there is so much available that some data can be used as a correlation for identification purposes.Users' data are also constantly in transit, being accessed by inside users and outside contractors, government agencies, and business partners sharing data for research. The collection and aggregation of massive quantities of heterogeneous data are now possible. Large-scale data sharing is becoming routine among scientists, clinicians, businesses, governmental agencies, and citizens. However, the tools and technologies that are being developed to manage these massive data sets are often not designed to incorporate adequate security or privacy measures.Privacy, for legal reasons, must be preserved even at the cost – not only monetary but that of system performance[14].

III. ARCHITECTURE

Map Reduce ,when the client request the data to the server the jobtracker will analyse the entire data and fetches the result. Map Reduce System (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 provide redundancy and fault tolerance mechanism.[13] In the default map reduce implementation system, intermediate data segments are pulled by Reduce Tasks in their entirety to local disk, then merged before being reduced for final results .Because of the physical movements of entire segments across disks we call this strategy as physical shuffling. .This is a time consuming process and whenever failure occurs the damage done should be within acceptable threshold rather than beginning the whole task.This also increases time complexity for accessing data from Map phase. Hence to overcome this process we propose a concept called as virtual shuffling [1].

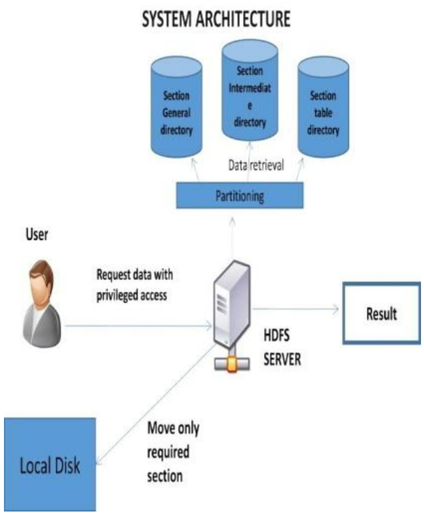


Fig2- System Architecture

Here first the User wants to create dropbox account and then only they are allowed to access the Network. Once the User creates an account, they need to login into their account and request the Job from the Service Provider. Based on the User's request, the Service Provider will process the User's requested Job and respond to them. All the User details will be stored in the Database of the Service Provider. In this Project, we will design the User Interface Frame to Communicate with the Server through Network Coding using the programming Languages like Java. By sending the request to Server Provider, the User can access the requested data if they authenticated by the Service Provider.Once the dropbox account has been created and the request is accepted,the user can register with his personal details.They can choose their designation either as controller,registrer or examiner.

In data generation, bulk of data will be generated, partitioned and uploaded in the cloud. Each STE(section table entry) in the STD(section table directory) represents the partial segment.Many STD's are organized into a section intermediate directory(SID),in section intermediate entry(SIE) represents an STD.Client request data to server with privileged access and chooses the region for retrieving data with minimum time frame.Once the user successfully signed in into the server, the user is requested to search their data based on query. In the query based, the data link will displayed as query.A server is a computer program running to serve the requests of other programs, the "clients". Thus, the "server" performs some computational task on behalf of "clients". The clients either run on the same computer or connect through the network. Here the Server acts as the main resource for the client.So the server will process the user's request and get the concerned data from the database.

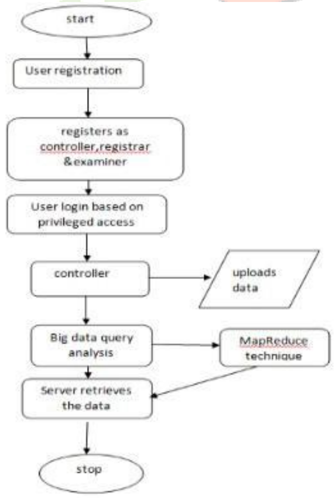
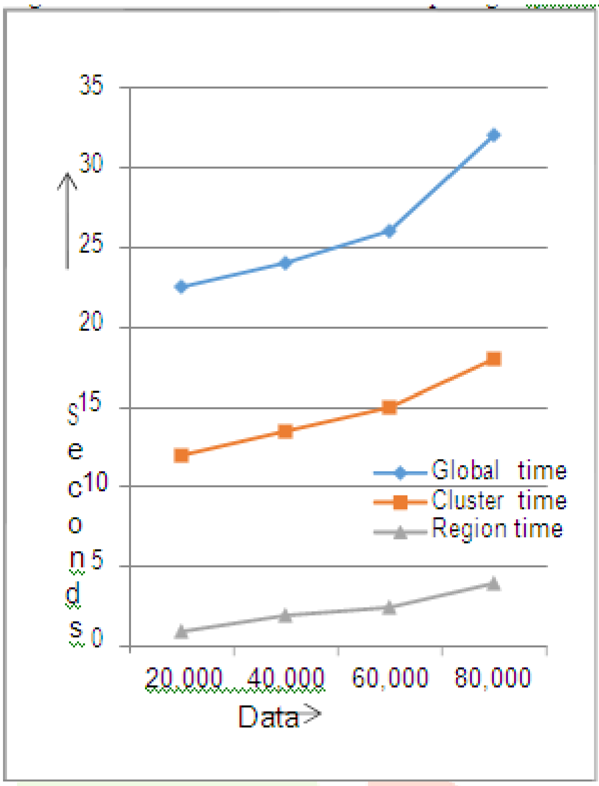


Fig 3- Dataflow diagram

Attribute-based encryption (ABE) can be used for log encryption. Instead of encrypting each part of a log with the keys of all recipients, it is possible to encrypt the log only with attributes which match recipients' attributes. This primitive can also be used for broadcast encryption in order to decrease the number of keys used.Once the search has been made, the result will be displayed to the user. From that displayed result, the user is requested to select the data that they are wanted.

IV. EXPERIMENTAL RESULTS

In this paper, we have proposed a strategy to speed up the process in map reduce along with a technique called as abstract based encryption. Virtual Shuffling reduces the time consumption. For data-intensive MapReduce programs such as TeraSort, data shuffling can add a significant number of disk accesses, contending for the limited I/O bandwidth. In order to elaborate this problem, we conduct a data-intensive MapReduce test, where we run 200 GB TeraSort on 10 slave nodes. We have examined the wait (queuing) time and the service time of I/O requests.



V. CONCLUSION AND FUTURE WORK

We have proposed shuffling as a new strategy to enable efficient data storage and retrieval for Map Reduce applications. Accordingly, we have designed and implemented a combination of three techniques including a three-level segment table, near-demand merging, and dynamic and balanced merging sub trees. Our experimental results show that shuffling strategy significantly relieves the disk I/O contention problem and speeds up data movement in Map Reduce programs. We efficiently handled fault tolerance mechanism and scalability issue. In addition, it reduces power consumption of Map Reduce programs by as much as 12%. For future work, we are interested in the applicability of virtual shuffling over different network protocols. We also plan to investigate the benefits of virtual shuffling for more commercial and scientific workloads on large-scale commercial cloud computing systems[1].

REFERENCES

[1] Weikuan Yu, Senior Member, IEEE, Yandong Wang, Xinyu Que, and Cong Xu, "Virtual Shuffling for Efficient computers, vol. 64, no. 2, February 2015.

[2] Debajyoti Mukhopadhyay, Chetan Agrawal, Devesh Maru, Pooja Yedale, Pranav Gadekar" Addressing NameNode Scalability Issue in HadoopDistributed File System using Cache Approach," 2014 13th International Conference on Information Technology.

[3] Lena Mashayekhy, Mahyar Movahed Nejad and Daniel Grosu, "Energy-Aware Scheduling of MapReduce Jobs for Big Data Applications,"in IEEE transactions on parallel and distributed systems, vol. 25,2015

[4] Yandong Wang, Xingu Que, Weikuan Yu, "Design and evaluation of network levitated merge for hadoop acceleration,"parallel and distributed system,IEEE transaction 2014,vol.25,issue:3,pages:602-611.

[5] Xuanhua Shi , Ming Chen , Ligang He ,Xu Xie,"Mammoth: Gearing Hadoop Towards Memory- Intensive MapReduceApplication," Parallel and distributed System,IEEE Transaction on (Volume:26, Issue:8),Page(s):2300 – 2315,Published in 2014

[6] Yanfeng Zhang, Shimin Chen, Qiang Wang, and Ge Yu"i2MapReduce: Incremental MapReduce for Mining Evolving Big Data," IEEE transactions on knowledge and data engineering, January 2014.

[7] PengHu and WeiDai, "Enhancing Fault Tolerance based on Hadoop Cluster," International Journal of Database Theory

and Application on 2014.
[8] Kenn slagper, Ching-hsien hsu, Yeh-ching chung, Daqiang zhang, "An improved partitioning mechanism for optimizing massive data analysis using Map Reduce ," published on 11th april 2013.
[9] https://en.wikipedia.org/wiki/Map-Reduce
[10]http://searchcloudcomputing.techtarget.com/definition/ Hadoop
[11] Book-"Pro Hadoop" by "Jason Venner" published by "APress" Publication.
[12] E-Book -"Map Reduce Parallel and Distributed programming model turorial pointssimple learning"
[13] E-Book "Big data and Hadoop" by V.K.Jain published by "Khanna Publication" page no: 229
[14] Lei Xu,Jian Wang"Information security in bigdata :Privacy and data mining" volume 2 IEEE access 2014