# FRAGDEG ALGORITHM FOR BIGDATA

Rekha and A. S.
Sathyabama University, Chennai, India
E-Mail: dulcetrekha@gmail.com

## ABSTRACT

The ultimate aim of this paper is to overcome the challenges faced in data mining and data warehousing in the field of big data. There are two types of data storage such as scalable and elastic. If it is scalable then existing techniques can be used. But while dealing with the elastic data, it needs concentration on many areas. It needs to concentrate on split up of data whenever the user adds some new data. It should be properly fetched without loss when needed no matter how many split-ups are there. Here a FRAGDEG Algorithm is used for integrating data. This fragmentation algorithm allows using a threshold value according to the user convenience. The big data handled in the existing best peer++ system provides platform for corporate network applications. This system delivers data sharing services for corporate networks with peer to peer data management platform. The total cost of ownership is reduced in inter companies. It eliminates the hadoop tool. The FRAGDEG algorithm works efficiently with bigdata on both velocity and variety aspect. The performance is made more efficient using this algorithm.

**Keywords:** big data, Hadoop, data mining, data warehousing.

## 1. INTRODUCTION

Cloud computing has turned into a standout amongst the most discussed innovations lately and has got loads of consideration from media and in addition examiners as a result of the opportunities it is putting forth.

Organizations of the same business segment are frequently associated into a corporate system for coordinated effort purposes. Every organization keeps up its own site and specifically imparts a bit of its business information with the others. Illustrations of such corporate systems incorporate production network systems where associations, for example, suppliers, producers, and retailers work together with one another to accomplish their particular business objectives including arranging generation line, making procurement methods and picking advertising arrangements.

From a specialized point of view, the key for the achievement of a corporate system is picking the right information offering stage, a framework which empowers the imparted information (put away and kept up by distinctive organizations) system wide unmistakable and backings productive explanatory questions over that information.

Customarily, information imparting is accomplished by building a brought together information distribution center, which occasionally extricates information from the interior generation frameworks (e.g., ERP) of every organization for consequent questioning. Sadly, such a warehousing arrangement has a few insufficiencies in genuine sending. First and foremost, the corporate system needs proportional up to bolster a large number of members, while the establishment of an expansive scale incorporated information stockroom framework involves nontrivial expenses including enormous equipment/programming ventures (aggregate expense of proprietorship) and high upkeep expense (aggregate expense of operations). In this present reality, most organizations are not quick to contribute intensely on extra data frameworks until they can obviously see the potential quantifiable profit (ROI). Second, organizations need to completely alter the entrance control strategy to figure out which business accomplices can see which a piece of their imparted information. Sadly, the greater part of the information distribution center arrangements neglect to offer such adaptabilities. At last, to boost the incomes, organizations regularly alterably modify their business methodology and may change their business accomplices. Accordingly, the members may join and leave the corporate systems voluntarily. The information distribution center arrangement has not been intended to handle such dynamicity.

In existence data sharing is done only for static data alone, with peer to peer data management. Best peer++ provides platform for corporate network applications. This system delivers data sharing services for corporate networks with peer to peer data management platform. The total cost of ownership is reduced in inter companies. It eliminates the hadoop tool but it is less efficient.

The main objective of this paper is to overcome the challenges faced in the field of big data and achieving elasticity. Depending on the data storage type, the proposed system "FRAGDEG Algorithm" can be used for extracting data concentrating on split up of data whenever the used add some new data. It should be properly fetched without loss when needed no matter how many split-ups are there.

## 2. RELATED WORK AND DIRECTIONS

C. Batini *et al* [1] clarifies the central standards of the database methodology are that a database permits a nonredundant, bound together representation of all information oversaw in an association. This is accomplished just when philosophies are accessible to bolster mix crosswise over hierarchical and application limits. The point of the paper is to give first a binding together structure to the issue of mapping combination, then a near audit of the work done so far around there. Such a system, with the related investigation of the current methodologies, gives a premise to distinguishing qualities and shortcomings of individual procedures, and general rules for future changes and expansions.

A. Abouzeid *et al* [2] examines the generation environment for investigative information administration applications is quickly evolving. Numerous endeavors are moving far from conveying their diagnostic databases on top of the line exclusive machines, and moving towards less expensive, lower-end, item equipment, normally organized in an imparted nothing MPP structural engineering, regularly in a virtualized situation inside open or private "mists". There have a tendency to be two schools of thought with respect to what innovation to use for information examination in such a situation. Advocates of parallel databases contend that the solid accentuation on execution and effectiveness of parallel databases makes them appropriate to perform such investigation. There have a tendency to be two schools of thought with respect to what innovation to use for information investigation in such a domain. Advocates of parallel databases contend that the solid accentuation on execution and effectiveness of parallel databases makes them appropriate to perform such investigation. Then again, others contend that MapReduce-based frameworks are more qualified because of their prevalent versatility, adaptation to non-critical failure, and adaptability to handle unstructured information.

B. Cooper *et al* [4] clarifies the idea of course heartiness for selecting the way in multi-jump system. Numerous channels are accessible on every connection. Utilizes an idea called sk While the utilization of MapReduce frameworks, (for example, Hadoop) for extensive scale information investigation has been generally perceived and mulled over, we have as of late seen a blast in the quantity of frameworks produced for cloud information serving. These more current frameworks location "cloud OLTP" applications, however they ordinarily don't bolster ACID exchanges. Illustrations of frameworks proposed for cloud serving utilization incorporate BigTable, PNUTS, Cassandra, HBase, Azure, CouchDB, SimpleDB, Voldemort, and numerous others. We display the "Yahoo! Cloud Serving Benchmark" (YCSB) structure, with the objective of encouraging execution correlations of the new era of cloud information serving frameworks.

## 3. PROPOSED SYSTEM ARCHITECTURE

Data Base server is created which manages the entire DB engine connected. The database will be stored in the DB engine via DB server. DB server registers the DB engine. DB server raise request to server for new DB engine. Since it is elastic model need for DB engine is non scalable.

Clients are created by registering them in the server. They will upload data in pay as you go model. They will retrieve their own data from the DB server. The data store in the DB engine will be in elastic model. The data of the particular user will be stored in various places. According to the space availability in the db engine the data of the user will be scattered and stored. When the data is retrieved then it will be integrated to provide as one file.

The data stored in DB engines will be mapped by the db server. When the user query for his data then the mapping of data will take place and then the joining of data will be done. The "n" number of split ups will be integrated and reduced to one particular file.

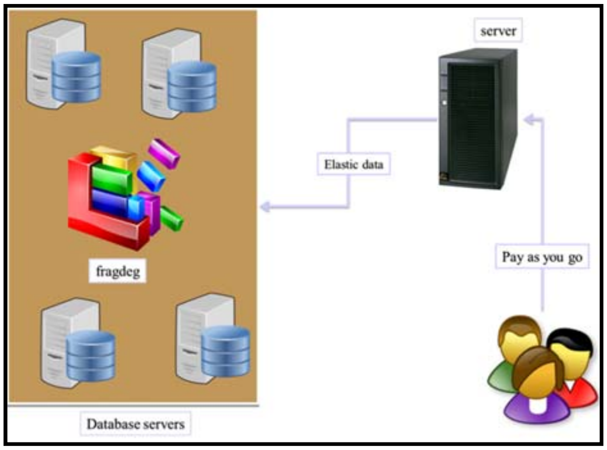

**Figure-1.** Proposed system architecture.

## 4. ALGORITHMS USED

The FRAGDEG algorithm does the collection of elastic data stored over the server racks. This collection happens over a fixed threshold time. When the threshold time value is reached the collection of data occurs and the storage is done in a single place. Through this instance we avoid various reference pointers for the data from a single user. Hence it eliminates the pointer movement between various storage racks. This algorithm runs on the back end without disturbing the current data storage by a user. Through this the performance level is always on a growing rate even if the data grows large. This can also be performed when the database is in idle state. The server can defragment all the db engines and likewise the individual DB engine can also do.

The FRAGDEG algorithm will improve data mining performance by dataware housing. While storing the data it will check for the free space according to the need. If the available free space is sufficient for the storage of all the split up of a particular user then it will defragment the files in one place. Through this FRAGDEG will improve performance for every particular period. This can also be performed when the database is in idle state. The server can defragment all the DB engines and likewise the individual DB engine can also do.

## 5. EXPERIMENTAL RESULTS

### 5.1 Experiment 1

First experiment is executed and by connecting the server and clients in a wired network through Wi-Fi modem. IP address was assigned statically. The clients were establishing network connection with the server and data were fed to different racks and fetched after using FRAGDEG threshold value and fetch the data without any loss efficiently.

### 5.2 Performance analysis

Fragmentation with sensitive attribute and association for exposed attributes in a fragment providing its consequences. For any visible yet confidential constraints having instance for data dependency conveys information about various attributes representing its correlation. Attributes for deriving implicit representation over the fragments enabling the link over cases have complex violations.

Implicit representation over computing the fragmentation exploits over data dependencies with its confidentiality constraints. Required design for new fragmentations approach for data dependency taken into consideration and handled easily. Extending the confidentiality for possible constraints can infer the cause for data dependency which can rewrite the constraints for every data dependency.

**Table-1.** Performance analysis of static vs elastic data.

| **Data retrieval speed (sec)** | **Time** (sec) | |
|---|---|---|
| | **Fragmentation with static data** | **Fragmentation with elastic data** |
| 2 | 0 | 0.5 |
| 4 | 0.2 | 0.5 |
| 6 | 0.3 | 1 |
| 8 | 1.2 | 3 |
| 10 | 1.8 | 4 |

In a static data implementation the storage space is wasted for a user without knowing whether the allocated space is used by the user efficiently or not. But when the case comes to dynamic data the usage of storage space is made more efficient by the usage of FRAGDEG algorithm. This algorithm integrates the data efficiently no matter how many split-ups are there and retrieves the data without any loss of data.

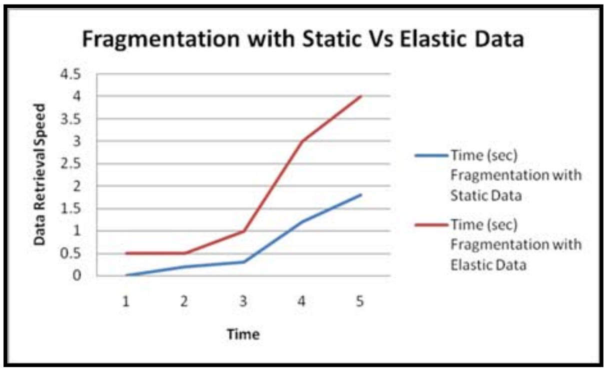

**Figure-2.** Performance analysis of static vs elastic data.

## 6. CONCLUSION AND FUTURE ENHANCEMENTS

The disadvantages in the existing system are overcome in the proposed system. The data on the server is fetched without any loss in a well efficient manner using the FRAGDEG algorithm. This is implemented for the elastic data. This can be used for multiple applications. This overcomes the usage of hadoop tool which is in the existing system. The big data handled in the existing best peer++ system provides platform for corporate network applications. This system delivers data sharing services for corporate networks with peer to peer data management platform. The total cost of ownership is reduced in inter companies. As it enhances Pay as you go model for efficient storage this model will be in existence soon.

The bigdata can be handled still more efficiently by this FRAGDEG algorithm. Further elastic data will come into existence. The variety of file types can be extended even for software files.

## REFERENCES

[1] C. Batini, M. Lenzerini and S. Navathe. 1986. A Comparative Analysis of Methodologies for Database Schema Integration. ACM Computing Surveys. 18(4): 323-364.

[2] D. Bermbach and S. Tai. 2011. Eventual Consistency: How soon is Eventual? An Evaluation of Amazon s3's Consistency Behavior. In: Proc. 6th Workshop Middleware Serv. Oriented Comput. (MW4SOC '11), pp. 1: 1-1: 6, NY, USA.

[3] A. Abouzeid, K. Bajda-Pawlikowski, D.J. Abadi, A. Rasin and A. Silberschatz. 2009. HadoopDB: An Architectural Hybrid of MapReduce and DBMS

Technologies for Analytical Workloads. Proc. VLDB Endowment. 2(1): 922-933.

[4] B. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan and R. Sears. 2010. Benchmarking Cloud Serving Systems with YCSB. Proc. First ACM Symp. Cloud Computing. pp. 143-154.

[5] V. Poosala and Y.E. Ioannidis. 1997. Selectivity Estimation without the Attribute Value Independence Assumption. Proc. 23rd Int’l Conf. Very Large Data Bases (VLDB ’97). pp. 486-495.

[6] E. Rahm and P. Bernstein. 2001. A Survey of Approaches to Automatic Schema Matching. The VLDB J. 10(4): 334-350.

[7] P. Rodrıguez-Gianolli, M. Garzetti, L. Jiang, A. Kementsietsidis, I. Kiringa, M. Masud, R.J. Miller and J. Mylopoulos. 2005. Data Sharing in the Hyperion Peer Database System. Proc. Int’l Conf. Very Large Data Bases. pp. 1291-1294.