International Journal of Research in Advent Technology (E-ISSN: 2321-9637) Special Issue
1st International Conference on Advent Trends in Engineering, Science and Technology
"ICATEST 2015", 08 March 2015

Secure Distributed Computation of Online Partitioned Data Clouds By MapReduce Framework

Asst. Professor Deepika Bhatia
Priyadarshini Bhagwati College of Engineering, Nagpur, deepika.18850@gmail.com, 09405350351

Abstract.
Distributed data mining (DDM) [1] is a process to extract globally interesting associations, classifiers, clusters, and other patterns from distributed data. As datasets double in size every year, moving the data repeatedly to distant CPUs brings about high communication cost. In this paper, we used data cloud to implement DDM in order to move the data rather than moving the entire computation. MapReduce is a software model for implementing data-centric distributed computing. Initially, a kind of cloud system architecture for DDM is proposed. After that, a modified MapReduce framework called pipelined MapReduce is presented.

Keywords: distributed data mining (DDM), Cloud Computing, MapReduce, Hadoop.

1. INTRODUCTION

Clustering is the process of discovering groups within high-dimensional databases, based on similarities, with a minimal knowledge of their structure. Traditional clustering algorithms perform it over centralized databases, however, recent applications require datasets distributed among several sites. Therefore, in distributed database environments, all distributed data must be concentrated on a central site before applying traditional algorithms.There is a series of limitations which creates problem in the utilization of traditional data mining techniques on distributed databases. The approach commonly taken, the gathering of all distributed databases in a central unit, followed by algorithm application, is strongly criticized, because in this case, it is important to take into consideration some issues, Namely: the possibility of existence of similar data with different names and formats, differences in data structures, and conflicts between one and another database. Besides, the unification of all of the registers in a single database may take to the loss of meaningful information, once that statistically interesting values in a local context may be ignored when gathered to other ones in a larger volume.
Other problems with centralized databases are:

1. It requires large data transference
2. Slow and costly process.
3. Addition or updation of new information creates problem. It needs very complex data updation strategy.
4. Furthermore, in some domains such as medical and business areas whereas distributed databases occurs, transferring raw datasets among parties can be insecure because confidential information can be obtained, putting in risk privacy preserving and security requirements.

Due to all of these problems related to database integration, research for algorithms that perform data mining in a distributed way is not recent. In the end of the 90s, several researches about algorithms to effectuate distributed data mining started to appear, having been strengthened mainly by the rise of the distributed database managing systems and need for analyzing this dispersed data.

2. RELATED WORK

a) Data partitioning methods

In cases in which the data set is unified and needs to be divided in subsets, due to its size, two approaches are normally used:

- **Horizontal Partitioning (Fig. 1a.)**
- **Vertical Partitioning (Fig. 1b.).**

The first approach deals with horizontally splitting database, creating homogeneous data subsets, so that each algorithm operates on different records considering, however, the same set of attributes. Another approach is vertically dividing the database, creating heterogeneous data subsets; in this case,

each algorithm operates on the same records, dealing, however, with different attributes.

	X1	X2	X3	X4
1				
-				
n-1				
n-2				
-				
p				

Fig. 1a. Horizontal partitioning

	X1	X2		X3		X4
1						
-						
n-1						
n-2						
-						
p						

Fig. 1 b.Vertical partitioning

If, on one hand, data partition methods keeps focusing on queries performance, seeking for the more suitable number of partitions to make the recovery process of stored data quicker, the presence of redundant or strongly correlated variables in a process of cluster analysis with self- Organizing maps, on the other hand, is not recommended. Therefore, in order to obtain better results in data analysis, the most recommended is geographically distributing data so that correlated variables stay in different units. Nonetheless in situations in which databases are already geographically distributed – not Self Organizing Maps - Applications and Novel Algorithm Design being possible to alter their structure – and the existence of strongly correlated structures may impair results, it is possible to utilize statistical techniques, such as Principal Components Analysis (PCA) or Factor Analysis to select a more suitable subset of variables and reduce these problems. Fig.2. shows the difference between Horizontal and vertical partitioning.

HORIZONTAL PARTITIONING(HP)	**VERTICAL PARTITIONING(VP)**
Horizontal partitioning is a model in which each processor holds a partial number of complete records of a particular table. It is more common in parallel relational database systems.	Vertical partitioning partitions the data vertically across all processors. Each processor has a full number of records of a particular table. This model is more common in distributed database systems
Different rows of a table at different sites	Different columns of a table at different sites
Advantages – Data stored close to where it is used ➔ efficiency Local access optimization ➔ better performance Only relevant data is available ➔ security Unions across partitions ➔ ease of query	Advantages – Data is geographically distributed so that correlated variables stay in different units(PCA is used to select more suitable subset of var.)➔ efficiency Local access optimization ➔ better performance Only relevant data is available ➔ security Unions across partitions ➔ ease of query
Disadvantages Accessing data across partitions ➔ inconsistent access speed No data replication ➔ backup vulnerability	Disadvantages Accessing data across partitions ➔ inconsistent access speed No data replication ➔ backup vulnerability Combining data across partitions is more difficult because it requires joins (instead of unions)
Number of queries is more.	No. of queries required is reduced.

Fig. 2. Difference between HP AND VP

b) Pipelined MapReduce Framework

In the data cloud system proposed in this paper, we studied an improvement in the traditional MapReduce framework, namely, Pipelined MapReduce Framework. In traditional MapReduce framework, the output of each Mapper is managed by the OutputCollector instance and stored in an in-memory buffer. The OutputCollector is also responsible for spilling this buffer to disk (i.e., HDFS in Hadoop) when the output reaches capacity of memory. The execution of a reduce task includes three phases: shuffle phase, sort phase and reduce phase. The shuffle phase fetches intermediate results from each Mapper. However, a reducer cannot fetch

the output of a mapper until JobTracker informs that the mapper is finished. The output produced by Reducers may be required by the next map step, which is also written to HDFS. There are at least two disadvantages within the traditional MapReduce framework.

HDFS should maintain enough storage for temporary files. Since each file has three copies in HDFS in default manner and most of DDM applications will produce a large amount of middle results, preserving storage for temporary files is extremely expensive and inefficient. Because Mapper and Reducer execute in a serial manner, and both Mappers and Reducers should spend plenty of time in reading middle data from HDFS, the execution speed of the traditional MapReduce framework is very slow. Fig 3 shows traditional and pipelined MapReduce.

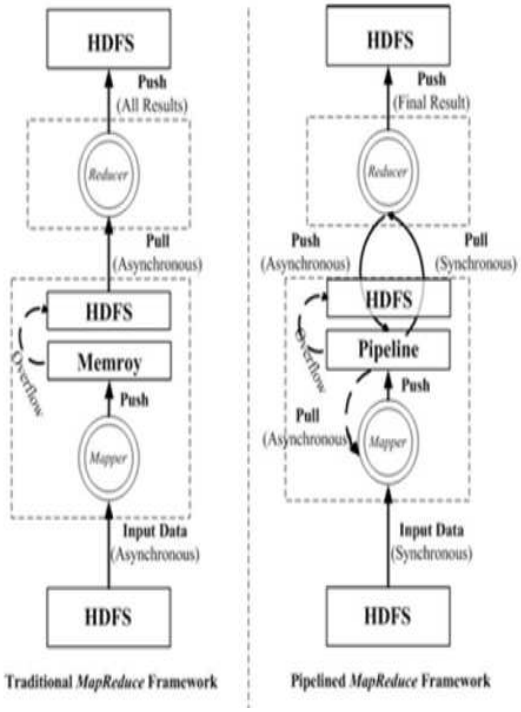

Fig 3 Traditional and Pipelined MapReduce.

HDFS should maintain enough storage for temporary files. Since each file has three copies in HDFS in default manner and most of DDM applications will produce a large amount of middle results, preserving storage for temporary files is extremely expensive and inefficient. Because Mapper and Reducer execute in a serial manner, and both Mappers and Reducers should spend plenty of time in reading middle data from HDFS, the execution speed of the traditional MapReduce framework is very slow.

To solve these above-mentioned disadvantages, pipeline is added between Mapper and Reducer. Middle data is stored in pipeline files and only final results are written to HDFS. Moreover, when the Mapper is executing, it simultaneously pushes middle data to Reducer via pipeline. Reducers then pull the middle data synchronously. Pipelined MapReduce Framework makes Mappers and Reducers execute in a parallel manner and also enhance the robustness of the fault tolerance.

3. PROPOSED APPROACH

Our main objectives are as follows:-

- To study Privacy preserving data mining.
- Secure multiparty communication and efficient anonymization of partitioned databases.
- Computing AND, OR operations etc. on data clouds.
- Online dynamic horizontal and vertical partitioning.
- Classification and clustering of data.
- Uploading of data using parallel MapReduce.
- Study of Data/Information loss during this process.

There are various issues to be dealt with in the paper as follows:-

- Privacy of data subjects using k-Anonymization.
- Privacy of data providers.
- Privacy of data providers using l-site diversity and l-diversity.
- To handle (key, value) pairs using MapReduce in efficient manner.
- No third party.
- Storing data in HDFS.
- Fault tolerance

- To increase effeciency and decrease cost.

We have proposed the following block diagram Fig.4, the architecure of secured distributed computation, for secured distributed computation of online data clouds using MapReduce framework technology. In this architecture, the data is uploaded online on the cloud from the original source locations either from medical data or any other private or public data sources. After this we apply horizontal or vertical data partitioning strategies. Classification and clustering [3] is applied on this partitioned data. HADOOP [2] is used for applying pipelined MapReduce architecture as discussed above using Self organizing maps [4], [5] and K-means clustering algorithms. After this various issues like Privacy of data providers using l-site diversity and l-diversity [6] are dealt with. Now we apply various computation like AND, OR computations on this distributed data, which will be more secure, efficient and reliable.

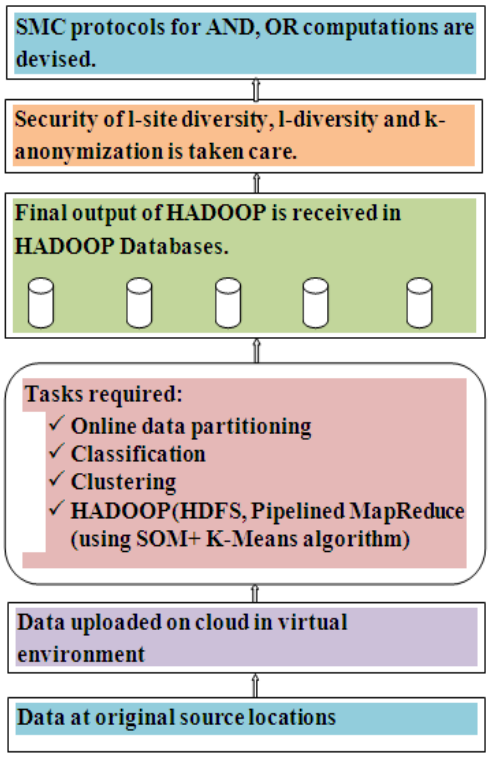

Fig.4 Architecure of secured distributed computation.

4. CONCLUSION

In the paper we have proposed a framework for secure data transfer on cloud platform. It used hadoop technology for secured transfer and to enhance the security we also used the parallel pipelined Mapreduce framework. It is more secured and more advanced technology. In future, we want to implement our module further for a particular database like hospital or banks etc.

REFERENCES

[1] Zhiang Wu, Jie Cao, and Changjian Fang Jiangsu ***Data Cloud for Distributed Data Mining via Pipelined MapReduce*** Provincial Key Laboratory of E-Business, Nanjing University of Finance and Economics, Nanjing, P.R. China zawuster@gmail.com, 2012.

[2] Hadoop: The Apache Software Foundation, http://hadoop.apache.org/core

[3] Ganesh P. et. Al. **Protection of Privacy in Distributed Databases using Clustering** International Journal of Modern Engineering Research (IJMER) www.ijmer.com Vol.2, Issue.4, July-Aug 2012 pp-1955-1957 ISSN: 2249-6645

[4] Gorgônio, F. L. & Costa, J. A. F. (2008b). Parallel Self-organizing Maps with Application in Clustering Distributed Data, *Proceedings of the International Joint Conference on Neural Networks*, IEEE World Congress on Computational Intelligence, Hong-Kong, 2008, Vol. 1, pp. 420.

[5] Gorgônio, F. L. (2009). *Uma Arquitetura para Análise de Agrupamentos sobre Bases de Dados Distribuídas*, Tese de Doutorado, Federal University of Rio Grande do Norte, UFRN/PPgEEC, Natal, RN, Brazil.

[6] Pawel Jurczyk and Li Xiong **Distributed Anonymization: Achieving Privacy for Both Data Subjects and Data Providers** pp. 191–207, 2009.IFIP International Federation for Information Processing 2009.