American Journal of Engineering Research (AJER)
e-ISSN : 2320-0847 p-ISSN : 2320-0936
Volume-03, Issue-06, pp-209-212
www.ajer.org

Research Paper Open Access

Compressed Data Transmission Among Nodes in BigData

Thirunavukarasu B[1], Sudhahar V M[1],VasanthaKumar U[1], Dr Kalaikumaran T[1], Dr Karthik S[1]

[1]*Department of computer science and Engineering, SNS College of Technology, India*

ABSTRACT: *Many organizations are now dealing with large amount of data. Traditionally they used relational data. But nowadays they are supposed to use structured and semi structured data. To work effectively these organizations uses virtualization, parallel processing in compression etc., out of which the compression is most effective one. The data transmission of high volume usually causes high transmission time. This compression of unstructured data is immediately done when the data is being transmitted from client to DataNode. Initially once unstructured or semi-structured data is ready for transmission, the data is compressed using some software tools or procedures. This compressed data is transmitted through certain medium that undertakes an effective transmission.*

Keywords–*BigData, Hadoop Architecture, Unstructured Data, Compression, Optimization, NameNode, DataNode, Data Transmission.*

I. INTRODUCTION

The corporation or organizations' success completely depends on how these corporations or organizations successfully manipulates or uses the vast amount of unstructured data. These unstructured data basically comes from website, XML files, Social Networks, etc. Some of common such examples includes Multimedia, web contents, satellite and medical contents.

1.1 Big Data

Big data is a large set of unstructured data even more than tera and peta bytes of data. Big Data[1] can be of only digital one. Data Analysis become more complicated because of their increased amount of unstructured or semi-structured data set. Predictions, analysis, requirements etc., are the main things that should be done using the unstructured big data. Big data is a combination of three v's those are namely Volume, Velocity and Variety. Big data will basically processed by the powerful computer. But due to some scalable properties of the computer, the processing gets limited.

1.2. Unstructured Data

Unstructured data is a data set those are in the form of logs. There won't be any items like rows, columns, records, etc., some unstructured data includes log details of website, multimedia contents, images, videos, satellite image contents, medical contents, etc. These unstructured data have become more complexes to be deal with. These unstructured data does not have any predefined data model. These unstructured data rather than having some text, will have vast amount of data like date, number, fact, etc., the unstructured data can never be readily classified. This kind of data cannot be contained in spreadsheet or relational database like structured data.

1.3. Compression

The compression[2] is a technique of reduction in size of large amount of structured or unstructured data. By using compression one can save some memory space and could be also able to minimize the transmission time. The compression could be made possible to be done for entre transmission unit or to certain data content.

By compression of data, the extra space characters can be removed. By introducing single repeat character for large repeated characters, substituting smaller bits can minimize the file up to 50% of its own contents. Algorithms can also be specified to determine how the compression should take place
Graphical data such as pictures, videos, animations, etc. are designed in such a way that it supports the data compression without any issues. The compression on these data can be of two kinds and they are

1 .Lossy compression
2. Lossless compression
In lossy compression, any certain information gets loss while the data is being compressed. In lossless compression there won't be any information loss during the data compression.

1.4. Need for Compression

Now relational database along with unstructured and semi-structured data evolves like xml, video, audio and images. These unstructured data are called large objects. Organizations are expected to deal with large amount of data volumes, any these kind of given data, the traditional data storage methods and tape can does not work anymore.

1.5. Compression Technique

- Huffman coding
- Arithmetic coding
- LZW compression
- LZ 77
- LZ 78, etc.

Among these compression techniques, the LZW[3] compression is favorable one with high performance ratio. It is an adaptive algorithm. Of the above mentioned methods, Huffman coding, Arithmetic coding are statistical methods. Whereas LZ (Lemple Ziv) algorithm is a dictionary method.

Here the basic idea used is replacing the recurring patterns with the references in the dictionary. In LZ Algorithm an explicit dictionary is maintained. The dictionary is primarily occupied with some set of data. Any further data could also be made to be added along with the dictionary. During the compression, initially every data is compared with the data in the dictionary. If the recurrence occurs, the original data is replaced the data n the dictionary. By this way the repeated data could be eliminated. Basically by LZ Algorithm, the previously processed data is used as dictionary.LZ 77 and LZ 78 are probably used to compress the data that consists of numbers and characters.

An advanced version of LZ Algorithm is LZW Algorithm. This algorithm is mostly used for the unstructured data like images, etc. But mostly used for the GIF images. Like other LZ algorithms, LZW is a patented one. By using LZ Algorithms, older entries are removed effectively.

II. HADOOP DATA TRANSMISSION TECHNIQUE

In existing methodology, the Hadoop architecture consists of master node or NameNode and DataNode[4]. The client uses the data from DataNode for the effective execution. Since BigData is a concept of distributed system. The replicas of data are made and placed in various Basic Blocks[5]. Certain set of Algorithms can be used for this replica placements.

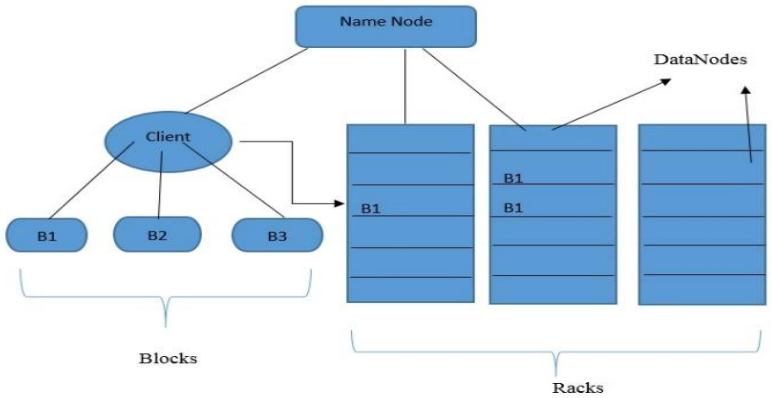

Fig 1. Data Transmission between Client and DataNodes

Initially the Large data set is divided into n number of blocks. These blocks are then replicated into some number of copies. Then, the blocks are placed in DataNodes as advised by the algorithm specified by the NameNode. The client then uses that DataNode. The Data set is transmitted to those DataNodes without any compression,

Node, the Mapping and Reducing is done for the execution of the data set. After once the execution gets completed, these data set with output is again send back to the client. This data sent was also a non-compressed one. The required analysis is made at the client side for business or organizational purposes. Here in this existing technique, it consumes more time for the transmission of the data. The block rocking algorithm which is an effective algorithm could be used for greater replica recovery purpose. The blocks thus created are arranged on the DataNode placing each replica of block in successive nodes

III. COMPRESSED DATA TRANSMISSION TECHNIQUE

In proposed method, the compression of unstructured data is made in effective manner. Initially the client, NameNode and DataNode are connected through some means of connection. Basically this connection is made through the TCP Connection. The TCP connection will be more advantageous as it always replies with a respond. The client receives the Acknowledgement from the DataNode about which DataNode should be used in the Rack. The client then as per the instructions given by the NameNode will effectively places the Blocks on the DataNode.

The Entire Dataset is at first spliced into basic blocks. The basic blocks are nothing but the splitting of the whole given dataset. These basic blocks are replicated into some number of copies. Then this Basic Block data are compressed by using a technique called LZ algorithm. Before performing the Compression, the basic Coupling of data set is made.

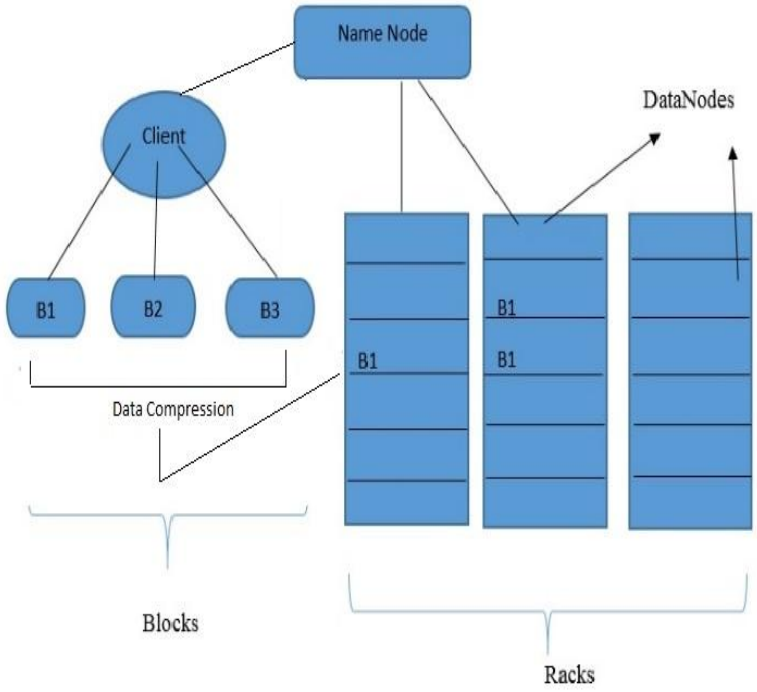

Fig 2.Compressed Data Transmission between Client and DataNode

3.1 Coupling

Coupling is nothing but grouping or classifying the dataset based on some criteria. Usually Coupling could be made based on the factors that includes size, type, etc. By Coupling, the dataset are initially grouped with certain similarities. Thus based on this grouping or classification, separate compression algorithms could be implemented for easy and elegant methods or Algorithms. For Graphical or Multimedia contents like Images, video, etc., an algorithm called LZW can be used in enrich able manner

3.2 Compression

The segregated data is at first made possibly grouped. Then the grouped similar kind of data is then again grouped. Once the data is grouped on some sort of similarities, the data Blocks are formed. Then on this data Blocks the compression is made. By this kind of compression a vast amount data could be eliminated from transmission. Since the data is considerably reduced, the transmission time obviously gets reduced. The compressed data then transmitted to the DataNodes. The Data is operated at the DataNode. In DataNode the Map Reduce Algorithm is executed. By map the data is again sub divided. Once the Sub division is made, the Map Algorithm will execute or will provide some sort of multi programmed technique to perform the required operation on those big set of data.

3.3 Decompression

Once the data is compressed and reaches the data node, there for the operations, the compressed data is again decompressed at DataNode level. By Decompression, the original Dataset is extracted, then the operation is performed on these original dataset. After the operations are performed, the data output is recovered. This recovered data is sent back to the client. The client receives this processed dataset. The Analysis on the data is basically made at this client side after all this steps are processed.

IV. MERITS AND DEMERITS

There are basically two kinds of compression methods. They are lossless and lossy compression. If the lossless compression method is used, the information after compression is extracted or decompressed without any data miss. But incase if the Lossy compression is prepared, then there will be some amount of data or information loss after the data extraction or decompression. Some advantages of compression includes the following,

- Reduction of cost
- Performance can be enhanced by achieving optimization.
- Great knowledge on which information the compression to take place.

V. RESULT AND CONCLUSION

Thus as a result, the large data set or BigData is transmitted after being performing the compression. Since data is completely compressed, the size gets reduced. As the Data size is reduced, it is much enough to transmit the lesser amount of data. This transmission of lesser amount of data achieves only a very low amount of transmission time. Thus the performance is enhanced. The speed of execution also gets increased, as the transmission time is reduced.

- Transmission time is inversely proportional to Amount of compression.
- Execution time is directly proportional to Transmission time
- Execution time is inversely proportional to Amount of compression

Generally the total performance of the Hadoop system gets increased. The optimization of the dataset is achieved by the enhanced Data Compression.

VI. ACKNOWLEDGMENT

I heart fully thank The Department of CSE, SNS College of Technology for the effective encouragement in achieving some milestones in BigData research. I also thank Dr. S N Subbramanian, Chairman - SNS College of Technology, for his endless support and guidance.

REFERENCES

[1]. Big Data Processing with Hadoop-MapReduce in Cloud Systems, Rabi Prasad Padhy,Senior Software Engg, Oracle Corp .,Bangalore, Karnataka, India.

[2]. J . Ziv and A. Lempel, "A Universal Algorithm for Sequential Data Compression," *IEEE Trans. On Information Theory*, pp. 337-343, May 1997.

[3]. J. Ziv and A. Lampel, "Compression of Individual sequence via variable-rate coding," *IEEE Trans. On Information Theory*, pp.530-536,sept. 1978 .

[4]. A. Wyner and J. Ziv, "The sliding window Lemple-Zi algorithm is asymptotically optimal," *Proc. IEEE*, pp. 872-877, June 1994.

[5]. B. Thirunavukarasu . Sangeetha K, kalaikumaran T, Karthik S, "Effectively Placing Block Replicas of Big data on the Rack by Implementing Block Racking Algorithm," *International Journal of Science, Engineering and Technology Research* , pp.891-894, April 2014