Learning Compression Approach Based On Euclidean Distance Algorithm For Big Sensing Data Processing On Cloud

Akanksha More
Department of Computer Engineering
Rajarshi Shahu College of Engineering
Pune,India
akankshamore91@gmail.com

ABSTRACT

We are living in the time of data explosion where big amount of data is generated from the different sources. The big sensing data is the data generating from the different sensing systems. This big sensing data is very important and necessary for the industry and for the scientific research applications. There are multiple sensing systems which generate the big amount of data. The main challenge is to process and store this big sensing data for future use. The storage cost of the big sensing data is too high because of memory requirement. Processing this data on the traditional database management system is hard complex and time-consuming. In this paper, we are proposing a method for processing and storage of the data. Here for the storage of the big sensing data we are compressing the big sensing data and storing it on the cloud which is very efficient. In this method, the big sensing data is partitioned into standard data chunks. The incoming big sensing data is compared to the generated standard data chunks using Euclidean distance similarity algorithm. After data chunks generation the chunks are compared by using compression algorithm. Here MapReduce algorithm is used for the compression. MapReduce algorithm provides extra scalability on the cloud. This proposed approach gives better data compression efficiency and loss of the data accuracy is also affordable.

Keywords— Big Sensing data, Cloud computing, Compression techniques, Euclidean distance algorithm, MapReduce algorithm for big data.

I. INTRODUCTION

Nowadays big sensing data plays important role in day to day life. The big sensing data is important for the industry and weather forecasting, scientific research, earthquake monitoring systems etc. There are different sources of the big sensing data such as sensing camera, video, satellites, meteorological sensors, traffic monitoring, and complex physics simulations [11].

The big sensing data is large in volume and it comes from the different sensing systems. Because of its large volume and high speed, it is very difficult to process it on the traditional database management system. Also, the storing of such a large amount of data in the memory is very expensive in terms of cost.

If for the processing some recursive algorithms are used for the big sensing data it can generate problems like memory bottlenecks and data deadlocks on data accessing. Because of these problems, we can use the cloud as the promising platform for the big sensing data processing and storage. On the cloud, we can store the big sensing data in affordable cost. But still storing the big sensing data on the cloud as it is was not a very efficient way. So before storing the data is processed i.e. Compressed and then it is stored efficiently. Cloud is efficient for processing and storing the big sensing data because of following reasons:

A. Powerful computational capability: Cloud has powerful computational capability than the traditional systems computational capability.

B. Storage: On a cloud, it is very efficient and affordable to store the big sensing data.

C. Resource reuse: We can reuse the resources on a cloud.

D. Low cost: It is very affordable to use the cloud for storage of the data.

There are many techniques for the processing of the big sensing data. But in the real world, generation rate and size of the big sensing data is very high therefore the current data processing methods need to be improved.

While processing big sensing data sometimes on-cloud computations can also generate some undesirable problems that can lead to unacceptable time cost and data processing failures.

To further increase data reduction and to decrease processing time cost we introduced a compression technique. In this method firstly the standard data chunks are generated. After the standard data chunks generation, the incoming big sensing data is compared to the data in standard data chunks. If the data is similar to the chunk data then it is not stored otherwise it is stored in the new chunk, for future data, it can be used as a reference data chunk. After the chunk generation, the chunks are compressed by using a compression algorithm. Here MapReduce algorithm is used for the extra stability on the cloud. The advantage of this compression technique is that while decompression we get entire chunk. So it is more convenient than the decompression of data unit approach.

II. REVIEW OF LITERATURE

In past decade, there are different systems and platforms are generated for processing and storing the big sensing data. Recently R. Buyya and C.S.Yeo [1] studies cloud computing and the different emerging platforms. It shows cloud as a market oriented resource allocation on cloud by the use of technologies like virtual machines. The study also shows market based resource management strategies which cover the service management and the service level agreement.

On cloud, there are different algorithms which increase the scalability of the cloud. B. Moseley and S. Suri [2], studies the MapReduce algorithm for the big data analysis. This describes the new technique in MapReduce framework called filtering. This technique is used for reducing the size of the input in the distributed fashion. In this process the input problems instance can be solved on the single machine. This also shows MapReduce can work for big data processing very effectively. Similarly C. Ji and Y. Li [3] study the big data processing on the cloud computing environment. They show different big data processing techniques based on the application approach. This shows MapReduce can be used for the parallel processing. Also shows the MapReduce parallel processing framework.

In this paper Euclidean distance similarity algorithm is used for the similarity calculation. W.Wang and D. Lu [4], shows the anomaly detection technique and the compression the data. They also show two fundamental technologies as distributed data store and complex event processing and the work flow description for distributed data processing.

Another similarity algorithm which can be used for the similarity is the Jaccard similarity algorithm. B. K. Samanthula and W. Jiang [5], used the Jaccard similarity for the multiset interaction. Jaccard coefficient is used as an information similarity measure. In addition of the similarity measure Jaccard has the advantage which protects the privacy of the data. They proposed the SJCM (Secure Computation of the Jaccard Coefficient for Multisets) using the existing dot product method. Processing big graph data on cloud is also a challenge. C. Yang and X. Zhang [6], shows a new technique for the processing and compressing the big graph data. In this technique big graph data is partitioned into the clusters. This experiment shows the spatiotemporal compression.

Cloud can provide the resources for storage and that storage is scalable and can be reusable. L. Wang and J. Zhan [7], studies how the storage of the scientific data. In this study different models are discussed like ESP (Enhanced Scientific Public cloud), HTC (High Throughout Computing) and MTC (Many Task Computing) etc. In all models cloud is used as the storage platform.

Although cloud is a good platform for storage there are many disadvantages for using cloud as a storage platform. S. Sakr and A. Liu [8], studies the different data management techniques. This study also gives the methods and approaches of deploying intensive data applications and also highlights some characteristics of the best candidate classes.

MapReduce algorithm is used for the scalability and for the parallel processing. Partitioning and parallel processing of the big data are main obstacles in the processing of the data. B. Li and E. Mazur [9], studies the cloud platform for the scalable one pass analytic using MapReduce algorithm on the Hadoop platform.

On cloud data processing is a complex process. Yahoo developed a method called Nova for the on cloud data processing. C. Olston and X. Wang [10], studies the Nova method. In this method the stream of incoming data is passed through the pig programs which are executing on the Hadoop structures.

While working on the cloud the privacy of the data is also considered W. Dou and X. Zhang [11], demonstrate the privacy aware cross-cloud service. They describes new method called Hiresome-II (History record based Service Optimization Method).In this technique the new services are selected on the basis of history record of that service. This Hiresome-II method based on the privacy of the cloud is protected.

In cloud computing the speed of the data processing and accuracy of data is important. N. Laptev and K. Zeng[12], studies the accuracy and the speed of the data processing on the cloud. For that they proposed EARL framework(Early Accurate Result Library).This method estimates error in the data processing.

L. Wang, Y. Zhang [13], studies the new method known as IMED[Image Euclidean Distance] which specializes into the spatial relationships of pixels. The used method IMED is applied to the image recognition. The Euclidean distance algorithm can be used very efficiently for the image recognition.

A. Strehl, J. Ghosh [14], studies different similarity algorithms are compared with each other. They study the clustering of the different web documents that can be used for semi-automated search. They also compare Euclidean distance similarity algorithm with Cosine similarity, Pearson correlation and extended Jaccard similarity algorithm.

S. Guha, R. Rastogi [15], ROCK [Robust hierarchical Clustering] method is used for the clustering of the data in data mining. This method uses the different approach for clustering than the traditional methods. It considers the links between the documents while the traditional approach mostly concentrated on the distance between the documents.

III. SYSTEM OVERVIEW

A. System description

The system works similar to the high frequent element compression but high frequent element compression only identifies the small data units.

In this paper for the partitioning of the data, 'data chunk' approach is used. The advantage of this method is that while decompression we will get the whole data chunk which is an efficient, time-saving and simple process.

In the previous approach after decompression, only the small data unit is obtained therefore that process is complex as compared to the data chunk compression method. Also, this compression identifies the complex data patterns while compression.

B. System Analysis

Initially, in this system the attributes of the big sensing data are studied. According to the attributes of the data, the complex patterns are decided.

These patterns and variations are first studied and defined. After standard reference chunks are generated.

Suppose big sensing data stream P which contains the element as x_i.

Let,

$P=(x_1, x_2, x_3,\ldots\ldots x_n)$

Where n=number of data units in P.

Here firstly the similarity between the data units is checked. Suppose we have to check the similarity between two data units x_1 and x_2 then function $sim(x_1,x_2)$ is used.

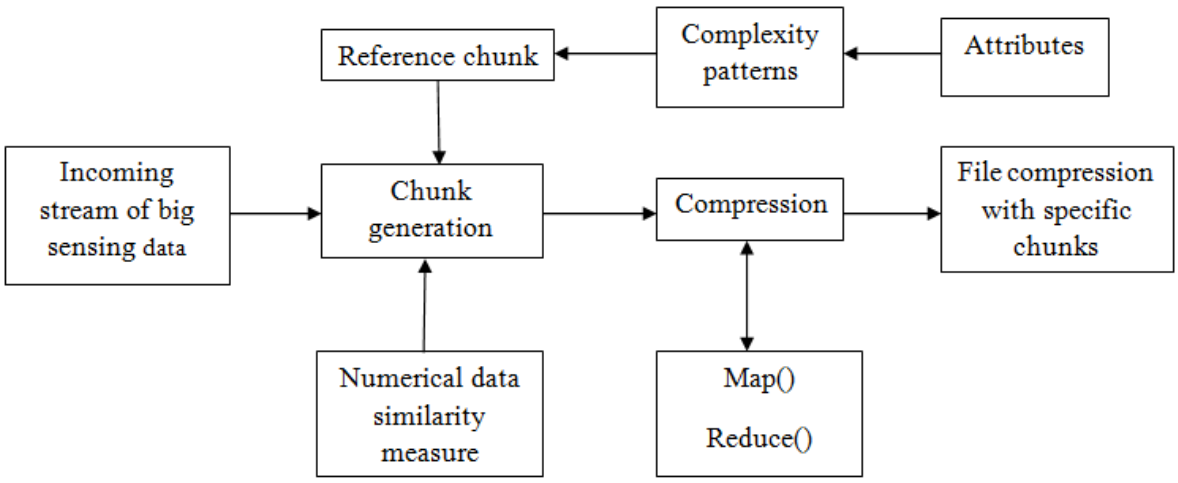

Fig. 1. SYSTEM ARCHITECTURE

For this similarity, the threshold value T is decided. If the similarity score is more than the threshold value then the data unit is taken as another standard reference chunk and if the similarity score is less than the threshold value then that data unit is decomposed in that set. So we can write

If $sim(x_1, x_2) > T$, then x_1 and $x_2$will be stored as different chunks and can be used as the reference chunks for the future incoming data.

If $sim(x_1,x_2) \leq T$, then only one data unit is stored. It can be as either x_1 or x_2. Here for calculating the similarity between data units the Euclidean distance algorithm is used.

After the standard reference data chunk generation, the incoming stream of the big data is compared with the standard data chunks which are created initially. After the comparison, chunks of the data are compressed by using the compression algorithm. The compression algorithm is embedded in the MapReduce algorithm for the extra scalability on the cloud. MapReduce algorithm is more strict therefore it cannot be used directly. So some functionalities of MapReduce are slightly changed. MapReduce has many advantages for using it for the big data and cloud. It provides data parallelism and data scalability.

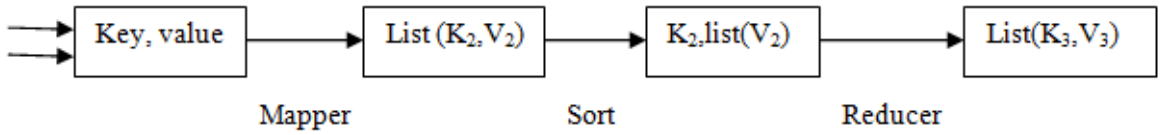

Fig. 2. Model of MapReduce algorithm

In this method, the compression algorithm is fixed into Mapper and Reducer side. The working programming model is shown in the given diagram above. Firstly the output of Mapper side is given to the Reducer as input. Therefore the compression algorithm is fixed in both Mapper side as well as Reducer side. In this way, the whole data chunk is compressed. This method can be classified into the following Modules.

Input file:

This is the first module of the system. Here the input is the stream of the big sensing data. After the input stream is given the input data is partitioned into chunks.

Chunk Generation:

In Introduction, we have represented an idea about chunk generation. For chunk generation, the streaming dataset is taken. Let the data units of the dataset are compared with each other by using the Euclidean distance algorithm. This compression method is same like compression of a high frequent element but the main difference is that the proposed method generates chunks after decompression and the high frequent element compression method generates simple data units after decompression. Another advantage of the proposed system is that the data chunk based compression identifies complex patterns while compressing the data which is beneficial for the processing of big sensing data.

Compression:

For the implementation of the proposed system we have to go through two important stages that are:

1. Generation of the data chunks.
2. Compression of the generated chunks.

In this module, we have used Map() and Reduce() method for compressing the data chunk.

C. Purpose

There are four purposes for this experiment

1. To increase the accuracy of the data.
2. To show the significant saving of storage space.
3. To show that only a small amount of data is lost during the compression process.
4. Saving of time and space requirement of data storage.

D. Mathematical Model

Input file set,

$P=(x_1,x_2, x_3,\ldots\ldots x_m)\ldots\ldots(1)$

m-number of data units streaming

Creation of standard chunk by the stream of big sensing data

$ch=(ch_1,ch_2,ch_3,\ldots\ldots ch_n)\ldots\ldots(2)$

ch-standard reference data chunk

n-number of data reference data chunks created.

Similarly of the streaming data with the generated reference chunk is checked by the Euclidean distance algorithm. If no similarity found then the data unit is stored as it is. Otherwise, the data unit is eliminated. After the similarity checking compression is carried out recursively.
Compression $\{ch_1,ch_2\}$
In compression, the MapReduce algorithm is used. In this process, the compression algorithm is embedded in the Mapper side as well as the reducer side.

E. Algorithms used

1) Standard data chunk generation algorithm

Input: The Stream of big sensing dataset P, maximum time threshold: r
Output: Data chunk set P' the subset of P
Process:

- In the initialization process starts with P' and its first element.
- Similarity mode is selected according to the application e.g. numerical data or text data.
- The y_1 is selected as a first element in the standard data chunk P'
- After adding y_1 the length of P' is set as 1.

2) Euclidean distance algorithm for finding similarity

Input: Generated reference data chunks, Stream of big sensing data.
Output: Classified data chunks
Process: The fix Threshold value T is taken.
Function Euclidean distance
distance=0
for dimension=1 to n
distance = distance + $(x_1[1] - x_2[2])^2$
next
return sqrt(distance)
end Function

Step 1: The distance between the two data units is calculated by this function.
Step 2: If the distance is greater than the threshold value T then the data units are not similar.
Step 3: If the data units are not similar then the data unit is stored as it is.
Step 4: If the distance between two data units is less than or equal to threshold value then the data unit is added in the chunk.

3) MapReduce algorithm for compressing the data chunks

The MapReduce algorithm is divided into two parts as Map() and Reduce()

1) Compression algorithm :Map():

A. Firstly the Map() takes P and P' as input.
B. The numerical data type is selected.
C. The total number of elements of the P' is measured and stored in L.
D. A Recursive similarity comparison function is called to check if there is an element in P which can be compressed and stored in P'
E. The elements which can be compressed are taken in the Map() function.

2) Compression algorithm :Reduce():

A. The Reduce() in the MapReduce algorithm extends tablereducer <>.
B. Reduce() function takes P as input
C. The compression mode is selected
D. The element P is compressed by using compress() function.
E. After the compression, the storage is updated.
F. An Index is stored for future decompression process.

IV. RESULTS AND DISCUSSION

In this paper, the accuracy of the Euclidean distance similarity algorithm and Cosine similarity algorithm is considered and compared for the big sensing data. These algorithms are applied to the big sensing data and accuracy and time factor considered for both algorithms. By these two factors, both algorithms are compared.

A. Performance Measure

The performance measures used are the space, time efficiency and accuracy of the system as compared to the existing system. The Cosine similarity algorithm is compared with the Euclidean distance algorithm in terms of accuracy, space and time efficiency.

B. Comparison of Cosine similarity and Euclidean distance similarity algorithm

In this paper, we are using a meteorological dataset which is the numerical data. Cosine similarity algorithm is used where we want to measure the length distance as well as angle distance. Euclidean distance is more reliable for numerical data.

C. Dataset

The meteorological data is used as a dataset. This is the big sensing data and it is in numerical format.

Here two similarity algorithms are compared to each other in terms of the accuracy of the data during the compression process. In this paper, we used the Euclidean distance algorithm for the similarity calculation between two data units. In this paper, we are using a compression algorithm which is embedded in the MapReduce algorithm. Here we are using Euclidean distance algorithm instead of Cosine similarity algorithm for similarity checking and comparing their results. By using this method the loss of data during the compression will reduce and it can be affordable as well as the decompression process will become easy because of chunk wise decompression. By this method, space and time are saved.

V. CONCLUSION

It is shown that the proposed compression based data chunk similarity improve the data compression the performance is also improved with less data accuracy loss. In comparison with the previous method, the compression ratio saves the space and time cost.

VI. FUTURE SCOPE

In future compression algorithm can be used which is based on the Spark for improvement in data processing. Spark has many advantages over the MapReduce algorithm. The Spark technology has a less expensive approach and can give better data processing results.

VII. ACKNOWLEDGMENT

We would like to thank the researchers as well as publishers for making their resources available and teachers for their guidance. We are thankful to our guide Mr. S. B. Javheri for their constant guidance. We are thankful to the authorities of Savitribai Phule University, Pune for their constant guidelines and support.

REFERENCES

[1] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg and I. Brandic, Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility Future Generation Computer Systems 25(6): 599-616, 2009.
[2] Shim, MapReduce Algorithms for Big Data Analysis, In Proc. of the VLDB Endowment, 5(12): 2016-2017, 2012.
[3] C. Ji, Y. Li, W. Qiu, U. Awada and K. Li, Big Data Processing in Cloud Environments, 2012 International Symposium on Pervasive Systems, Algorithms and Networks, 2012, pp. 17-23.
[4] W. Wang, D. Lu, X. Zhou, B. Zhang and J. Wu, Statistical Wavelet-based Anomaly Detection in Big Data with Compressive Sensing, EURASIP Journal on Wireless Communication and Networking, 2013.
[5] Bharath K. Samanthula and Wei Jiang, Secure Multiset Intersection Cardinality and its Application to Jaccard Coefficient IEEE Transactions on Dependable and Secure Computing.
[6] C. Yang, X. Zhang, C. Liu, J. Pei, K. Rama mohanarao and J.Chen, A Spatiotemporal Compression based Approach for Efficient Big Data Processing on Cloud, Journal of Computer and System Sciences (JCSS). vol. 80: 1563-1583, 2014.
[7] L. Wang, J. Zhan, W. Shi and Y. Liang, In cloud, can scientific communities benefit from the economies of scale? IEEE Transactions on Parallel and Distributed Systems 23(2): 296-303, 2012.
[8] S. Sakr, A. Liu, D. Batista, and M. Alomari, A survey of large scale data management approaches in cloud environments, Communications Surveys and Tutorials, IEEE, 13(3): 311336, 2011.
[9] B. Li, E. Mazur, Y. Diao, A. McGregor and P. Shenoy, A platform for scalable one-pass analytics using mapreduce, in: Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD11), 2011, pp. 985-996.
[10] C. Olston, G. Chiou, L. Chitnis, F. Liu, Y. Han, M. Larsson, A. Neumann, V.B.N. Rao, V. Sankarasubramanian, S. Seth, C. Tian, T. ZiCornell and X. Wang, Nova: Continuous pig/hadoop workflows, Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD11), pp. 1081-1090, 2011.
[11] W. Dou, X. Zhang, J. Liu and J. Chen, HireSome-II: Towards Privacy- Aware Cross-Cloud Service Composition for Big Data Applications, IEEE Transactions on Parallel and Distributed Systems, 26(2): 455-466, 2015.
[12] N. Laptev, K. Zeng and C. Zaniolo, Very fast estimation for result and accuracy of big data analytics: The EARL system, Proceedings of the 29th IEEE International Conference on Data Engineering (ICDE), pp. 1296-1299, 2013.
[13]Liwei Wang, Yan Zhang, Jufu Feng, On the Euclidean Distance of Images,Center for Information Sciences School of Electronics Engineering and Computer Sciences,100871,2000.
[14] Alexander Strehl, Joydeep Ghosh, and Raymond Mooney, Impact of Similarity Measures on Web-page Clustering, The University of Texas, Austin, TX, 78712-1084,2000.
[15] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim, ROCK: A Robust Clustering Algorithm For Categorical Attributes, Advanced Institute of Science and Technology and Advanced Information Technology Research, 2000.