International Journal of Computer Science Engineering & Technology
APPROVED BY
NATIONAL SCIENCE LIBRARY (NSL)
NATIONAL INSTITUTE OF SCIENCE COMMUNICATION AND INFORMATION RESOURCES (NISCAIR)
COUNCIL OF SCIENTIFIC AND INDUSTRIAL RESEARCH (CSIR)- NEW DELHI INDIA
ISSN :2455-9091

ANALYSIS FOR BIG SENSING DATA IN CLOUD COMPRESSION MODELS USING CHUNKING AND DIFFERENT COMPRESSION APPROACHES

[1] UshaRani S,
Research Scholar,
[1] Mother Therasa Women's University, TamilNadu, India.

[2] Dr.K.Kungumaraj
[2] Assistant Professor,
[2] Arulmigu Palaniandavar Arts College for Women, TamilNadu, India.

ABSTRACT

Big data is linked with the entireties of composite data sets. In big data environment, data is in the form of unstructured data and may contain number of duplicate copies of same data. To manage such a complex unstructured data hadoop is to be used. Current big sensing data processing on Cloud have adopted some data compression techniques. However, due to the high volume and velocity of big sensing data, traditional data compression techniques lack sufficient efficiency and scalability for data processing. Based on specific on-Cloud data compression requirements, we propose a novel scalable data compression approach based on calculating similarity among the partitioned data chunks. The main objective of the paper is to design a load rebalancing algorithm to reallocate file chunks such that the chunks can be distributed to the system as uniformly as possible while reducing the movement cost as much as possible. In this method firstly the big data is divided into the different chunks and then these chunks are compressed according to the chunks. Here the another algorithm known as Jaccard and Map reduce is used to find the similarity between the data. Then this algorithm is compared with the existing systems cosine similarity algorithm.

Keywords: [Big data Compression, chunking, Map reduce, Jaccard, Cosine, Similarity]

1. INTRODUCTION

In today's modern civilization everything or device is digitized. No one can survive without this digital space. This digitization creates lots of data in Exabyte. This data is in the form of unstructured or redundant data. The main issues in big data were that how to

maintain this large amount of data, who maintains it and where to store this big data. Indeed, even today Facebook, twitter and other long range social communication site creates extensive measure of information in consistently. In this way, there is extensive degree on big data. As there are many methods and advances to break down the information. There are different methods to investigate the big data i.e Data de-duplication techniques. As when the information is expanding then there are more possibilities that information is rehashed or excess that necessities to prerequisite of huge stockpiling devices. It additionally expands the preparing time of the query. So there is an extensive degree to break down the information by information de-duplication systems with the goal that can decrease the necessity of expansive data and diminish the handling time to execute the problem. The process of elimination of related data or duplicate data is called de-duplication process. It is divided into three parts data based, location based and disk based.

Cloud is a promising platform for big data processing with its powerful computational capability, storage, resource reuse and low cost processing the big sensing data is still costly in terms of space and time. For the reduction of the time and space cost of big data different techniques should be needed. If some recursive algorithms are used to process the big sensing data it can produce many problems such as memory bottlenecks, deadlocks on data accessing. In this paper the firstly the similarity data chunks are generated. Then the similarity algorithm is applied on these data analysis which shows significant speed up. Here the Jaccard similarity algorithm is used. In this paper we are comparing the both algorithms on the factor accuracy of the data. The accuracy and the space cost parameters are directly proportional to each other i.e. as the accuracy of the data increases the more space is required to store that data. The similarity algorithm Jaccard is used to find the data similarity more accurately than Cosine similarity algorithm. In results we will compare both the algorithms i.e. Jaccard algorithm and cosine similarity algorithms on the basis of the accuracy of the data preserved i.e. how accurately they finds the similarity of the data. As compared to the Cosine similarity algorithm the Jaccard algorithm is more advantageous in some measures as given. After finding the similarity by Jaccard algorithm the algorithm is applied to the data chunks is known as MapReduce algorithm. The MapReduce algorithm is used is for the compression of the similar data chunks. A dynamic deduplication approach has been used where in which Content Defined Chunking (CDC) is used to create variable size chunks which defines the breakpoints with the use of Basic Sliding Window (BSW). Frequency Based Chunking (FBC) identifies the frequently obtained chunks by following a two phase process, one is chunk frequency estimation using an estimation algorithm and second is two stage chunking in which coarse grain and then fine grain chunks are obtained using CDC.

2. LITERATURE SURVEY

Zhi Tang, Youjip Won proposed made a Content Based File Chunking model structure which melded a CPU chunking subsystem and GPGPU subsystem. **Zhike Zhang, Zejun Jiang, Zhiqiang Liu, Cheng Zhang Peng** proposed a novel chunk-based de-duplication methodology which could stay away from a summary in RAM and in this way evade all RAM use basic to hold an once-over. They kept up various canisters in plate, which basically contained distinctive chunk IDs. **Duane F. Shell, Leen-Kiat Soh, Vlad Chiriacescu** concentrated on indicating chunking and its effect on recovery, learning, and adequacy. **Xingyu Zhang, Jian Zhang** displayed de-duplication gathering. The Approach of joining comparability with area was related with the de-duplication gathering. **Wen Xia, Hong Jiang, Dan Feng, Lei Tian** introduced DARE, a de-duplication cautious, low overhead likeness disclosure and end plot for delta weight on the most essential reason for de-duplication on stronghold datasets. **Bo Mao, Hong Jiang, Suzhen Wu, Lei Tian** proposed POD, an execution orchestrated de-duplication arrangement, to update the execution of central stockpiling structures in the Cloud by utilizing information de-duplication on the I/O way to deal with evacuate excess make demands while moreover sparing storage room. **W. Wang, D. Lu, X. Zhou, B. Zhang and J. Wu** introduces the two fundamental technologies is the distributed data store and another is complex event processing and shows the work flow description for distributed data processing. **Bharath K. Samanthula and Wei Jiang** proposed 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 protocal(Secure computation of the Jaccard Coefficient for Multisets)using the existing dot product method. **C. Yang, X. Zhang, C. Liu, J. Pei, K. Rama mohanarao and J.Chen** shows the definition of the anomaly detection and the big data. The anomaly detection is based on the uncompressed data due to storage burden and the inadequacy of privacy protection and compressive sensing theory introduced and used in the anomaly detection algorithm. This anomaly detection technique used for the through-wall human detection to demonstrate the effectiveness. **L. Wang, J. Zhan, W. Shi and Y. Liang** consists of following folds:1)scientific communities which are small to medium scale utilize the elastic resources on the public cloud site while maintaining their flexible system control. 2)Lightweight service management for making management tasks simple and service heterogeneous workloads. **S. Sakr, A. Liu, D. Batista, and M. Alomari** gives the approaches and mechanisms of deploying intensive data applications which are gaining scalability, consistency, economical processing of large scale data on the cloud and also highlights some characteristics of the best candidate classes.

3. ANALYSIS ON DIFFERENT APPROACHES

Describe the architecture and its key features followed by the detailed discussion of its design and working. The architecture is designed to improve the Data elimination ratio and storage space in a globally based storage system.

The architecture consists of three functional modules named as: Pre-processor module, Commonality Detector module, and Pcompressor module, in addition with the three key features namely as Metadata Server, Index Server, Backup Client which are defined below:

1. Pre-processor: The pre-processor performs the very first step of a de-duplication process by dividing the input data into chunks. It uses rabin based CDC for the formation of chunks and generation of their hash values.

2. Commonality Detector: In this module, there are three key features namely Index Server, Metadata Server and Backup Client.

- Metadata Server: The hash values of chunks generated by pre-processor are stored at metadata server. The stored values are further being forwarded to the index server for comparison of hash index.
- Index Server: The hash values of input data are then compared with the already stored hash values in the index at index server for duplicate detection. If similar data is found then it is redundant. If not, the index will be updated. The Frequency and Commonality detector detects the most frequent chunk occurring which is to be stored at backup client for further use if needed.
- Backup Client: The backup Client stores the most frequently occurring chunks. The tiger hash is employed to generate hash values because of its fast processing. On the basis of generated hash values the redundant and non-redundant data is separated.

3. Pcompressor: The Pcompressor employ Pcompress utility at the redundant data. The Pcompress utility provides much meaningful and useful data after compression.

In proposed work main algorithm is used top process big sensing data. So, some features of big sensing data will be studied and analysed. To carry out compression, the similarity between two different data chunks should be defined. So, how to define and model the similarity between data chunks is a primary requirement for data compression. After the definition for the above similarity model for data chunks, how to generate those standard data chunks for future data compression is also a critical technique which we designed. A novel compression algorithm is developed and designed based on our similarity model and standard data chunk generation.

Input File This is the first module of the system. Here user gives the file as input. User can input any formatted file as input. Once user inputs the file will transmitted for generating chunks in next module.

Chunk Generation Here we have introduced technique for generating chunks of data. In the introduction we have represented an idea about data chunk which is based on compression. This is theme is not useful for compressing the data. It is like compression of high frequent element. Here difference is that compression of these elements identifies only simple data units. But our data chunk based compression identifies complex partition and pattern during compression process. Similarity algorithms for finding the similarity between the data chunks.

Here the Jaccard similarity algorithm is used to finding the similarity between the data chunks and the data stream. In existing system the cosine similarity algorithm is used for finding the similarity. Following the both algorithms are given. 1) Similarity model for the cosine similarity algorithm: The cosine similarity works for both text data and as well as text data. In this paper we are taking the data which is of weather forecasting and in the form of numerical format only. Therefore we are using cosine similarity algorithm only for the numerical data. In cosine similarity algorithm the cosine angle between the two vectors is calculated. If the angle between the two vectors is zero then the similarity between vectors is one. For the larger value of angle the similarity is less. 2) Similarity model for Jaccard similarity algorithm: Jaccard Similarity measure is another measure for calculating the similarity. In this algorithm, the index starts with a minimum value of 0 (completely dissimilar data) and goes to a maximum value of 1 (completely similar data). For implementing proposed system on cloud we need to go through two essential stages that are generation of data chunk and compression based on these chunks. That is why two algorithms have been developed respectively for data processing in these two stages given above. In the this module we have used Map (), and Reduce() method for compressing the data chunk. With above algorithms and techniques used in the proposed system, we will get the compressed files with specific chunks.

3.1 APPROACHES USED

1) STANDARD DATA CHUNKS GENERATION ALGORITHM

Input:

Streaming big sensing data set S;maximum time threshold for chunk evolvement:r

Output:

Data chunk set S’ which is a subset of S

Process:

Initialize process is conducted including S0 and its first element

Similarity mode is calculated and selected according to application requirement e.g.numerical data

The first element x1 in big data set selected as a first element in the standard data chunks set S'

Length of the S' is set as 1.

2) JACCARD ALGORITHM FOR FINDING SIMILARITY

Input:

Generated data chunck, Data

Process

The fix value Threshold T is taken.

The data is compared to that T value.

If the value is less than the T then the data is added to the chunk.

If the value is greater than T then the data is stored as it is.

The similarity can be find out by the Jaccard function.

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. Mapper side takes S and S' as input.

B. The numerical data type is selected.

C. The total numbers of elements in the standard data chunk S ' is calculated and stored in L.

D. Recursive similarity comparison function is called again to tag any data element in S to find any x S which could be compressed.

E. The data elements which could be compressed are tagged in map() function.

2. Compression algorithm:Reduce(): A. The reducer side compression algorithm extends the tablereducer <> of mapreduce model.

B. Reduce() function takes S as input

C. The compression model is selected

D. The element S is compressed by using compress() function.

E. After the compression the storage is updated.

F. Index is stored for future decompression process

4. ANALYSIS RESULTS

In this paper the accuracy of the two similarity algorithms is considered and the compared for the big sensing data. The algorithms are applied to the database and the accuracy of the both algorithms are to be compared.

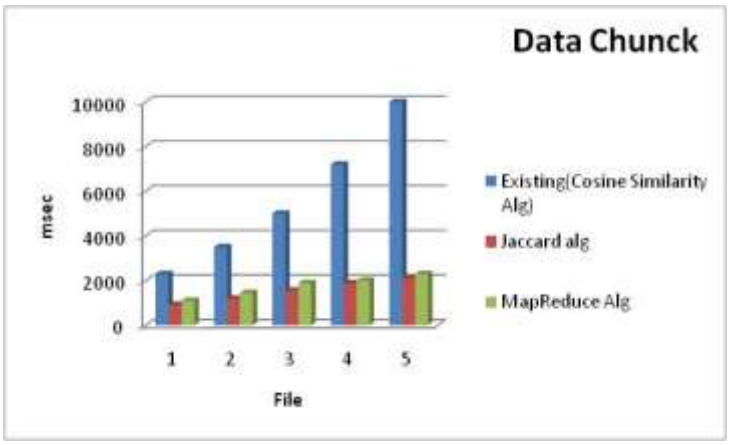

Figure 1: Data Chunk

Figure 1 gives the comparison of the Jaccard similarity algorithm, Map Reduce algorithm and the Cosine similarity algorithm. Here the time factor is considered for the comparison of the algorithms. As shown in the figure different sizes of data chunks are considered and the timestamp is compared.

Here three similarity algorithms are compared to each other in terms of accuracy of the data during the compression process. In this paper we used Jaccard similarity algorithm and Map Reduce algorithm. We use the numerical meteorological numerical dataset. The data is divided into the data chunks depending upon similarity of the data. For similarity the Jaccard algorithm is used and for compression Map Reduce algorithm is used. This will increase the accuracy of the data.

CONCLUSION

Big data contains data in unstructured form and have redundant data. To manage this unstructured data and reduce duplicity from data various chunking techniques and de-duplication techniques has been used. It is demonstrated that our proposed scalable compression based on data chunk similarity improve the data compression performance gains with data accuracy loss. The significant compression ratio brought which is space and time cost saving. In future the compression algorithm can be used which is based on the Spark for improvement in data processing. Spark has many advantages over the map-reduce algorithm.

The Spark technology has less expensive approach. With capabilities like in- memory data storage and near real-time processing, the performance can be several times faster than other big data technologies.

REFERENCES

[1] Zhi Tang, Youjip Won, "Multithread Content Based File Chunking System in CPU GPGPU Heterogeneous Architecture", 2011 First International Conference on Data Compression, Communications and Processing, China, pp. 58-64, 2011.

[2] Zhike Zhang, Zejun Jiang, Zhiqiang Liu, Cheng Zhang Peng, "LHS: A Nobel Method Of Information Retrieval Avoiding An Index Using Linear Hashing With Key Groups In Deduplication", Proceedings of the 2012 International Conference on Machine Learning and Cybernetics, China, pp.1312-1318, 2012.

[3] Duane F. Shell, Leen-Kiat Soh, Vlad Chiriacescu, "Modeling Chunking Effects on Learning and Performance using the Computational-Unified Learning Model (C-ULM): A Multiagent Cognitive Process Model", IEEE 15th International Conference on Cognitive Informatics & Cognitive Computing, India, pp. 77-85, 2016.

[4] Xingyu Zhang, Jian Zhang, "Data Deduplication Cluster Based on Similarity- Locality Approach", IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, CA, pp.2168-2173, 2013.

[5] Wen Xia, Hong Jiang, Dan Feng, Lei Tian, "Combining Deduplication and Delta Compression to Achieve Low-Overhead Data Reduction on Backup Datasets", Data Compression Conference, France, pp. 203-212, 2014.

[6] Bo Mao, Hong Jiang, Suzhen Wu, Lei Tian, "Leveraging Data Deduplication to Improve the Performance of Primary Storage Systems in the Cloud", IEEE Transactions on Computers, NY, pp.1- 14, 2015.

[7] 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.

[8] Bharath K. Samanthula and Wei Jiang, Secure Multiset Intersection Cardinality and its Application to Jaccard Coefficient IEEE Transactions on Dependable and Secure Computing.

[9] C. Yang, X. Zhang, C. Liu, J. Pei, K. Rama mohanarao and J.Chen, A Spatiotem-poral Compression based Approach for Efficient Big Data Processing on Cloud, Journal of Computer and System Sciences (JCSS). vol. 80: 1563-1583, 2014.

[10] L. Wang, J. Zhan, W. Shi and Y. Liang, In cloud, can scientific commu-nities benefit from the economies of scale? IEEE Transactions on Parallel and Distributed Systems 23(2): 296-303, 2012.

[11] 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.

[12] 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.

[13] C. Olston, G. Chiou, L. Chitnis, F. Liu, Y. Han, M. Larsson, A. Neu-mann, 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.

[14] 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.

[15] N. Laptev, K. Zeng and C. Zaniolo, Very fast estimation for result and accuracy f big data analytics: The EARL system, Proceedings of the 29th IEEE International Conference on Data Engineering (ICDE), pp. 1296-1299, 2013.