DATA CHUNK SIMILARITY BASED COMPRESSION APPROACH FOR BIG SENSING DATA IN CLOUD

Seljo Jose[1] | T.B.Dharmaraj[2]

[1](UG Student, Christ The King Engineering College, seljoalapattu@gmail.com)

[2](Assistant Professor, Christ The King Engineering College, bellidharmaraj@gmail.com)

Abstract— *Cloud computing provides a promising platform for big sensing data processing and storage as it provides a flexible stack of massive computing, storage, and software services in a scalable manner. 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 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. First process is to allocate the chunks of files as uniformly as possible among the nodes such that no node manages an excessive number of chunks*

Keywords— Big Sensing Data; Cloud Computing; Data Chunk;Data Compression; Similarity Model; Scalability; Load Balancing

1. INTRODUCTION

IT is becoming a practical requirement that we need to process big data from multiple sensing systems. That is, we enter into the time of data explosion which brings about new scientific challenges for big sensing data processing. In general, big data is a collection of data sets so large and complex that it becomes extremely difficult to process with on-hand database management systems or traditional data processing tools. It represents the progress of the human cognitive processes, usually includes data sets with sizes beyond the ability of current technology, method and theory to capture, manage and process the data within a tolerable elapsed time According to literature since 1980s, generated data doubles its size in every 40 months all over the world. In the year of 2012, there were 2.5 quintillion (2.5 _ 1018) bytes of data being generated every day Hence, how to process big data has become a fundamental and critical challenge for modern society.

A very important source of big data is sensing systems, including camera, video, satellite, meteorology, connect omics, earthquake monitoring, traffic monitoring, complex physics simulations, genomics, biological study, medical research, gene analysis and environmental research The big sensing data from different kinds of sensing systems is high heterogeneous, and it has typical characteristics of common real world big data. They are five 'V's, Volume, Variety, Velocity, Veracity and Veracity.

To overcome the processing difficulties caused by five 'V's, of big sensing data, the trend to deploy big data processing on Cloud is getting popular day by day. Cloud computing provides a promising platform for big data processing with its powerful computation capability, storage, scalability, resource reuse and low cost, and has attracted significant attention in alignment with big data. In Amazon's recent real world big data processing on Cloud projects most of big data sets come from sensing systems. However, to process big sensing data can still be costly in terms of space and time even on Cloud platform. To reduce the overall time and space cost for big data, especially big sensing data processing on Cloud, different techniques have been proposed and developed But due to the size and speed of big sensing data in real world, the current data compression and reduction techniques still need to be improved. It has been well recognized that big sensing data or big data sets from mesh networks such as sensor systems and social networks can take the form of big graph data. To process those big graph data, current techniques normally introduce complex and multiple iterations. Iterations and recursive algorithms may cause computation problems such as parallel memory bottlenecks, deadlocks on data accessing, algorithm inefficiency .In other words, under some circumstances, even with Cloud platform, the task of big data processing may introduce unacceptable time cost, or even lead to processing failures. To further improve the data size reduction, reduce the processing time cost and release the iterations in processing big sensing data, in this paper, we propose a novel technique based on data chunk partitioning for effectively processing big data, especially streaming big sensing data on Cloud. With this novel technique, big sensing data stream will be filtered to form standard data chunks at first based on our pre-defined similarity model. Then, the coming sensing data stream will be compressed according to generated standard data chunk

With the above data compression, we aim to improve the data compression efficiency by avoiding traditional compression based on each data unit, which is space and time costly due to low level data traverse and manipulation. At the same time, because the compression happens at a higher data chunk level, it reduces the chance for introducing too much usage of iteration and recursion which prove to be main trouble in processing big graph data.

The contents of this paper are organized as follows. In Section 2, we review related work and conduct problem analysis. In Section 3, a data similarity model will be defined and introduced. With that similarity model, the formation process of standard data chunks will be offered by training initial data stream. Then, we will introduce our streaming sensing data compression according to the standard data chunks. In Section 4, all the related scalable algorithms are offered, including scalability with Mapreduce, standard data chunk generation algorithm and scalable compression algorithm. In Section 5, the experimental results will be analyzed to show significant data compression performance gains. In addition, the accuracy loss will also be discussed in relation to compression effectiveness. , we will conclude the paper with a brief outlook of future work

2. RELATED WORKS

Some techniques have been proposed to process big data with traditional data processing tools such as database, traditional compression, machine learning, or parallel and distributed system In the following Section 2.1, those current popular techniques for big data processing on Cloud will be introduced and analyzed.

Nowadays, lots of big data sets or streams come from sensing systems which are widely deployed in almost every corner of our real world to assist our everyday life In order to cope with that huge volume big sensing data, different techniques can have been developed on-line or off-line, centralized or distributed. Naturally, the computational power of Cloud comes into the sight of scientist for big sensing data processing

Cloud computing provides comprehensive computing and storage resources enabling a pay-as-you-go business model by offering IT resources as services As a result, Cloud provides a promising scalable platform for big data storage, dissemination and interpreting At present, some research has been done about how to process big data with Cloud. For example, Amazon EC2 infrastructure as a service is a typical Cloud based distributed system for big data processing. Amazon S3 supports distributed data storage. MapReduce is adopted as a programming model for big data processing with Cloud. MapReduce has been widely revised from a batch processing framework into a more incremental one for analyzing huge-volume of incremental data on cloud. It can sort petabytes of big data in only a few hours. The parallelism also provides some possibility of recovering from partial failure of servers or storage during the operation. In our work, MapReduce also acts as a base for parallel processing on Cloud.

A significant amount of research has been done on the processing of incremental data on cloud. Kienzler et al. developed a “stream- as-you-go” approach for accessing and processing incremental big sensing data on cloud via a stream based data management architecture. The extension of traditional Hadoop framework was made to develop a novel framework named Incoop by incorporating several techniques like task partition and memorization-aware schedule. Olston et al. present a continuous workflow system called Nova on top of Pig/Hadoop through incremental data processing.

Sensor-Cloud is a unique sensing data storage, visualization and remote management platform that leverages powerful cloud computing technologies to provide excellent data scalability, fast visualization, and user programmable analysis. Sensor-Cloud platform has been developed including its definition, architecture, and applications. However, the Sensor-Cloud has less consideration for the big data in complex network topology. Due to the features of high variety, volume, and velocity, big data is difficult to process using on-hand database management tools or traditional Sensor-Cloud platform. The typical examples of big sensing data of complex networks are social network and large scale sensor networks. Under the theme of those complex network systems, it may be difficult to develop time-efficient detecting or trouble-shooting methods for big data processing in complex network systems in real time Current typical techniques such as MapReduce may introduce high computation cost when encountering big graph sensing data. More work is still expected to improve the effectiveness and efficiency in terms of big graph data processing on Cloud. Therefore, we aim to offer an optimal solution for real-time streaming big graph data compression for applications on Cloud.

Specifically, to reduce the volume of big data sets, different data reduction methods have been proposed recently. For example, in paper the work considers compressed sensing for sparse and low- rank tensors. Low-rank tensors were synthesized as sums of outer products of sparse loading vectors, and a special class of linear dimensionality-reducing transformations that reduce each mode individually. It was proved that interesting “oracle” properties exist. The proofs naturally suggest a two-step approach for processing big sensing data on Cloud. However, the extension and improvement are required to face new issues of big data and Cloud.

In paper a data quality (DQ)-centric big data infra-structure for federated sensor service clouds was proposed. The paper explores the advantages and limitations of current big data technologies in the con-text of Cloud-enabled large scale sensor networks, which naturally complement the emerging big sensing data paradigm. It focuses in particular on the issue of representing and managing Big Data, with emphasis on analytics over Big Data, as well as processes and architectures working with such data, with emphasis on Wireless Sensor Networks (WSNs), and draws future directions in this field. In paper Recovery algorithms are developed in compressive sampling (CS). Specifically, to speed up the least-squares module, the matrix-inverse-update algorithm is adopted. That developed algorithm has the potential to be used for compressing big sensing data on Cloud. But cannot be used directly due to the new requirement such as extreme high data speed, distributed environment and scalability. In paper an anomaly detection technique was used for through-wall human detection to demonstrate the big sensing data processing effectiveness. This technique is totally based on compressive sensing. The results showed that the proposed anomaly detection algorithm could effectively detect the existence of a human being through compressed signals and uncompressed data .In paper an adaptive data gathering scheme by compressive sensing for wireless sensor networks was developed. By introducing autoregressive (AR) model into the reconstruction of the sensed data, the local correlation in sensed data is exploited and thus local adaptive sparsity is achieved.

3. SCALABLE ALGORITHMS FOR DATA CHUNK SIMILARITY BASED COMPRESSION ON CLOUD

With the generated standard data chunks set S0 , the scalable compression algorithm based on MapReduce programming model is offered as follows. The algorithm is divided into two components including Mapper side compression algorithm and Reducer side compression algorithm. First, we introduce our Mapper side algorithm.

“Map()” Side Algorithm. Scalable Compression with Data Chunk Similarity

```
public static class Mapper extends Table Mapper <. .. . ., . . .. . .> {
public Mapper() {}
@Override
public Datatype map(Datatype S ¼ fx1; x2; . . . ; xng, Data- type S ¼ fv1; v2; . . . vkg.)
throws IOException {
ImmutableBytesWritable value ¼ null;
initialize X; // a temporary variable for storing recursively selection from fxi; . . . ; xjg;
if(mode.equal(mumerical_data))
Compression.set(Simn1(,), Simn2(,));
if(mode.equal(text_data))
Compression.set(Simn1(,), Simn2(,), Q);
Compression.set(Simn1(,), Simn2(,), Q, SimE(,), SimV(,));
L ¼ MaxElementSizeof(S); int start ¼ 0;
for(; S! ¼ ø; start ¼ start þ L){
X ¼ S.getlement(start, L);
for(int j ¼ L; j>0; j–){
C(vj, X);
tag(X.Distance<Threshold);
}
}
return S; // a tagged data set S for final compression;
0 for (int i ¼ 0; i < S
try {
context.write(compressionID,value);
} catch (InterruptedException e) {
throw new IOException(e);
}
}
}
```

two important stages, standard chunk generation and chunk based compression are essential. So the algorithms are developed respectively to conduct the related data processing for the above two stages.

At the first stage, the standard data chunks are generated. The algorithm for selecting those chunks can be performed before the real data compression by centralized computer systems. So, a centralized algorithm is offered for describing the whole process of standard data chunk generation. At the second stage of big data compression, the storage and time saving is mainly achieved by chunk based compression and scalability of Cloud. The chunk based compression is introduced by the algorithm itself, and the scalability is introduced by designing the compression algorithm with MapReduce. In other words, the compression algorithms conclude two parts, “Mapper” side algorithm and “Reducer” side algorithm. In following content of this Section 4, all the above algorithms will be offered and analysed To guarantee the scalability of the proposed data compression algorithm based on data chunks, MapReduce programming model and Hadoop 4 4

4. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a novel scalable data compression based on similarity calculation among the partitioned data chunks with Cloud computing. A similarity model was developed ton compressing big data sets. Instead of compression over basic data units, the compression was conducted over partitioned data chunks. The MapReduce programming model was adopted for the algorithms implementation to achieve some extra scalability on Cloud. With the real meteorological big sensing data experiments on our U-Cloud platform, it was demonstrated that our proposed scalable compression based on data chunk similarity significantly improved data compression performance gains with affordable data accuracy loss. The significant compression ratio brought dramatic space and time cost savings. With the popularity of Spark and its specialty in processing streaming big data set, in future we will explore the way to implement our compression algorithm based on data chunks similarity with Spark for better data processing achievements.

REFERENCES

[1] S. Tsuchiya, Y. Sakamoto, Y. Tsuchimoto, and V. Lee, “Big data processing [2]A. Cuzzocrea, G. Fortino, and O. Rana, “Managing data and processes in cloud-enabled large-scale sensor networks: State-of-the-art and future research directions,” in Proc. 13th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput., 2013, pp. 583–588.

[2] Y. Fang, L. Chen, J. Wu, and B. Huang, “GPU implementation of orthogonal matching pursuit for compressive sensing,” in Proc. 17th IEEE Int. Conf. Parallel Distrib. Syst., 2011, pp. 1044–1047.

[3] W. Wang, D. Lu, X. Zhou, B. Zhang, and J. Wu, “Statistical wave-let-based anomaly detection in big data with compressive sensing,” EURASIP J. Wireless Commun. Netw., 2013, Doi: 10.1186/ 1687-1499-2013-269.

[4] J. Wang, S. Tang, B. Yin, and X. Li, “Data gathering in wireless sensor networks through intelligent compressive sensing,” in Proc. IEEE INFOCOM, Mar. 2012, pp. 603–611.

[5] S. H. Yoon and C. Shahabi, “An experimental study of the effectiveness of clustered aggregation (CAG) leveraging spatial and temporal correlations in wireless sensor networks,” ACM Trans. Sens. Netw., vol. 3, no. 1, Art. no. 3, 2007.

[6] R. Qiu and M. Wicks, “Cognitive networked sensing and big data,” ISBN 978–1–4614–4544—9, DOI 10.1007/978–1–4614–4544–9.

[7] Real Time Big Data Processing with Grid Gain (2017, Feb. 16). [Online]. Available: http://www.gridgain.com/sitemap/

[8] Managing and Mining Billion-Node Garphs (2017, Feb. 16). [Online]. Available: http://kdd2012.sigkdd.org/sites/images/ summer school/Haixun-Wang.pdf

[9] Hadoop (2017, Feb. 16). [Online]. Available: http://hadoop. apache.org