Volume 8, No. 5, May-June 2017
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

A Fast Block-level Error Identification and Rectification of Error Burst in Big data on Cloud

Vinita Nareda
Department of CSE
Yagyavalkya Instituite of Technology
Rajasthan Technical University, India

Ankur Goyal
Professor and Head of Department of CSE
Yagyavalkya Instituite of Technology
Rajasthan Technical University, India

Abstract: Big data is a heap of energetically increasing data sets that needs to be effectively seized, accumulated, allocated and examined to pull out useful information. The problems of storing, assessing, security and safety, and analysis are altogether amplified by the speed, volume, and diversity of big data. Wireless sensor systems and networks are an important source of big data generating a wide variety of huge data sets. Data transmitted among sensor nodes is generally prone to noise and interference resulting in errors caused by failure in nodes and wireless communication. Reliability is the fundamental necessity of any communication. Effective identification and rectification of sensor big data errors on cloud is a challenging issue. The proposed error identification approach is specifically brought for detecting errors in big data sets of sensor networks. The proposed approach is based on assembling the datasets of multiple nodes of WSN into a single group called blocks, thereby making groups of datasets of nodes in a given system. Grouping the nodes permits to define a set of nodes that can be thought as a "single node". It eliminates the computation overhead of locating errors at each node, thereby identifying and correcting the errors at block level. To justify that our results are efficient, comparison graphs are generated for time efficiency, accuracy and performance.

Keyword: Burst error, Big data, Block level error code, Cloud computing, Error identification and rectification.

1. INTRODUCTION

Big data is a heap of energetically increasing data sets so massive and complicated that its processing has become a fundamental and crucial problem. Cloud computing is very significant in Big data analytics because of its distributed computing that includes computing, storage, networking and analytical software and cost effective properties.

Cloud computing delivers a variable load of enormous computing, repository, and software solutions in an extensible way at a small cost. Its facility to spend as-you-utilize the computing services on a limited basis and release them as required, makes it very ideal for data processing. Big data sets collecting from different sources like WSN usually gets corrupted and lossy due to wireless communication and presence of many hardware faults and inaccuracies in the nodes. For a given application to deduce perfect and appropriate result, it is necessary that the data gathered is faultless, clean, accurate, and lossless. However, effective identification and cleaning of big data errors is a very challenging issue which requires new solutions. Security is a prime and challenging issue for the data that is deployed. When it is a matter of processing of large data sets, Hadoop's MapReduce programming permits for the processing of such large volumes of data in a completely safe and cost-effective manner. In this paper, a rapid and effective spatial-temporal procedure for error identification is proposed to locate error bursts in big data sets on clouds thereby improving network performance. The importance of proposed approach is to preserve integrity of data and significant time improvement.

Intent of the proposed paper given as:

1. Node Side error location and revision.
2. Network Performance enhancement.

2. RELATED WORK

Cloud computing has been an important technology in business as well as in education. Cloud computing is a model that concentrates on sharing data and its computations in an extensible network consisting of nodes[1]. Hadoop is known as an open source java framework, a workaholic for the cloud computing[2]. Hadoop's distributed file system and its map reduce can handle abundant data. Various nodes in a cluster can be evaluated in parallel with the help of powerful Hadoop Distributed FileSystem (HDFS) and its Java-based APIs. HDFS provides a master and slave organization. Namenode represents a master node and data nodes represent slaves[3].Namenode handles the information about the file system[4]. HDFS loads the file-system as a sequence of blocks, the size of every block is reserved 64MB by default. In case data loss happens, HDFS copies the data for 3 times to give well-made storage.

The distributed file system of map reduces divides the data on many different machines and data is depicted in (key, value) form of pairs. On one machine called as master, the MapReduce implements the main function where the input data is processed before calling the map functions and afterwards the output of reduce function is processed[5]. The MapReduce programming model consists of two functions: The map function changes the input data into key and value pairs, and then reduce function is applied for every list of values that match to that key. This model provides the users a fast use of resources by removing complex systems problems. The MapReduce system typically applies "group data according to key", then it applies the reduce function to each group, to attain parallelism. The model allows parallelism as (key, value) pairs can be extracted in parallel to the reduce function to every group on different nodes. This is known as MapReduce group by model.

Wang et al.[6] presented different errors in networks that involve six different categories of standard errors with misplaced or incorrect data. It achieves a good foundation for evolving various methods for finding errors for social networks. It acts as a basis for error detection and correction in social networks. Yang[7] suggested a categorization for errors introduced in big data sets and then for fast error detection and correction, the network feature of a clustered network is analyzed. But, in this approach, issues like big data cleaning and recovery needs to be improved.

Mukhopadhyay[8] gave a model based error rectification method for WSN. This technique is based on rectification with data trend prediction. Because the work[8] is fast error detection by intelligent sensors, its processing capability and time performance are extremely limited when encountering big data sets. Sheth[9] develops a decentralized fault diagnosis system for WSN. It allows efficient management of a WSN by diagnosing the true root cause of low performance by combining multiple sensor observations. This requires minimum data collection at the centralized base station. Qin[10] developed a new cloud application that manages Big Data in a distributed environment while operating at an appropriate high bandwidth speed using Map Reduce. In case of failure of more than one node, Qin illustrated an appropriate joint rectification through simple regenerative high bandwidth codes and MDS codes in the cloud.

3. PROPOSED WORK

We include two particular basis: Hadoop for error handling and Ubuntu for relentless prepare.
MapReduce programming model provides processing for huge data sets in an extensively parallel manner. We propose a block level error identification and rectification to eliminate the computation overhead introduced by processing each line in the whole process of error localization. In our proposed solution, we introduce coding as well as error identification and rectification at block level in place of node level as done in the existing work. Existing solution generates codes and error localization for each node i.e. for each line of memory so it takes a lot of time to scan and fix each line in case of error rectification. Grouping multiple nodes into block of data makes it more efficient as an entire block of data can be replaced at a time in case of error identification. So, block based replacement is fast and more efficient.

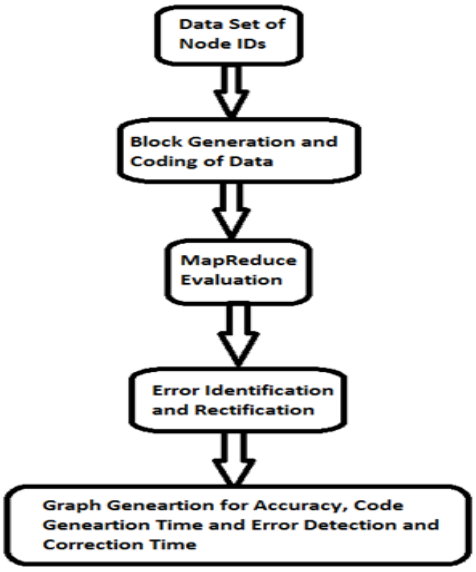

Figure. 1 Coding & Error Identification and Rectification procedure for a given Big Data size

4. PROPOSED WORK

Run the Ubuntu platform and start the hadoop application as shown in Figure.2.
/startprogram.sh

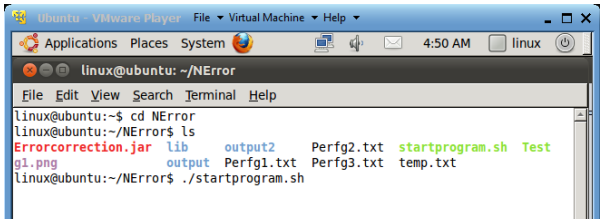

Figure. 2 Application initialization on Ubuntu

Next, figure shows the appending the input file containing big data to create the blocks of data by grouping it and then encoding the data.

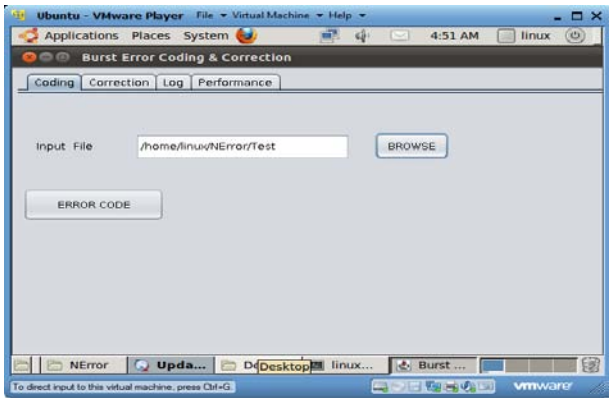

Figure.3 Appending of input file to detect and correct errors.

The input file big data is grouped in blocks and then encoded to generate a file containing blocks of data named as blockcontent file as shown in Figure.4.

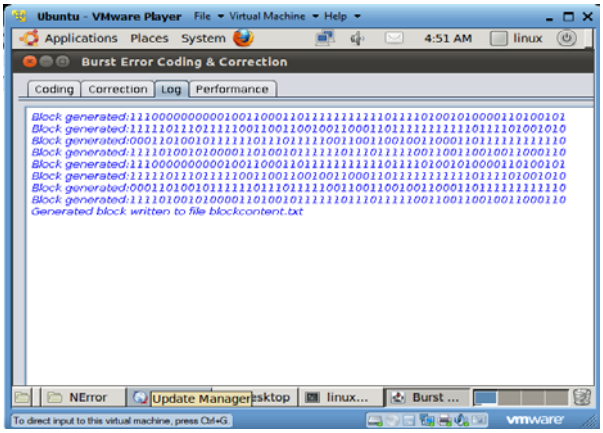

Figure.4 Block content file

Now, we append the blockcontent file so as to identify and rectify any error introduced in this file as shown in Figure.5.

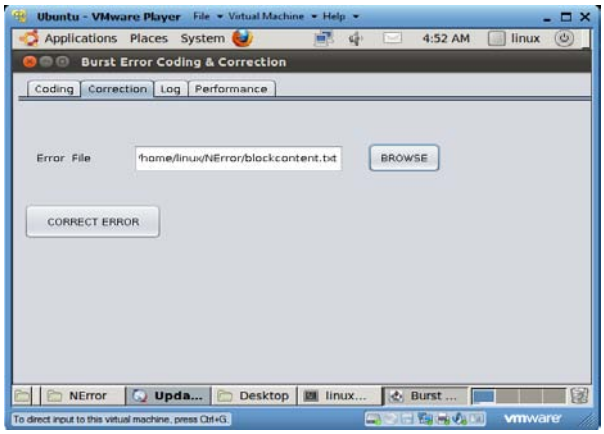

Figure.5 Appending blockcontent file

In case, any error is found in the blockcontent file, it is detected as well as corrected as shown in Figure.6.

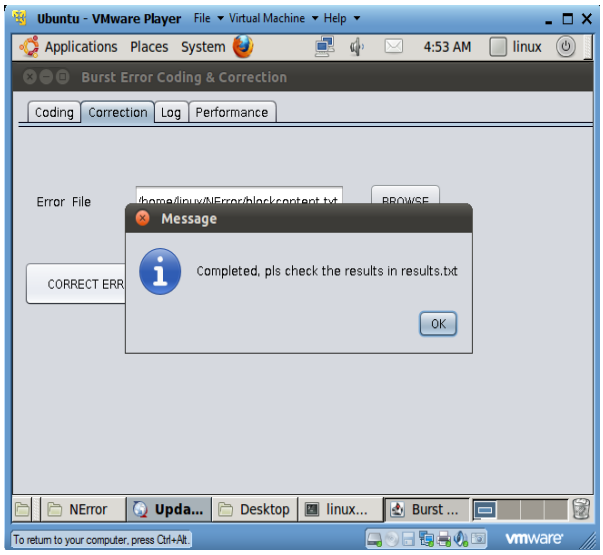

Figure.6 Error identification and rectification

Like this way, computation overhead at each line (node) is eliminated and an efficient spatial-temporal block level error identification approach is proposed.

5. RESULTS

The results include the comparison of outputs generating in terms of graphs for accuracy of code generation, code generation time and error correction time are as follows.

Table 1: Accuracy calculation for different inputs files size

S.no	Size of Input file	Existing Accuracy%	Proposed Accuracy%
1	20	89.62	70.0
2	40	89.33	65.0
3	60	88.86	66.6
4	80	87.16	67.5
5	100	84.92	66.9
6	120	81.90	77.2

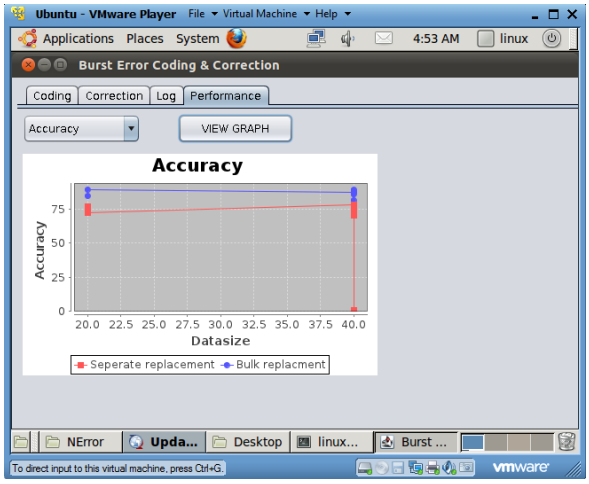

Figure.7 Graph for accuracy of code generation

Table 2: Code generation time calculation for different inputs file size

S.no	Size of Input File	Existing Code Genertion Time(ms)	Proposed Code Generation Time(ms)
1	20	7	3
2	40	20	7
3	60	37	21
4	80	43	26
5	100	67	33
6	120	61	32

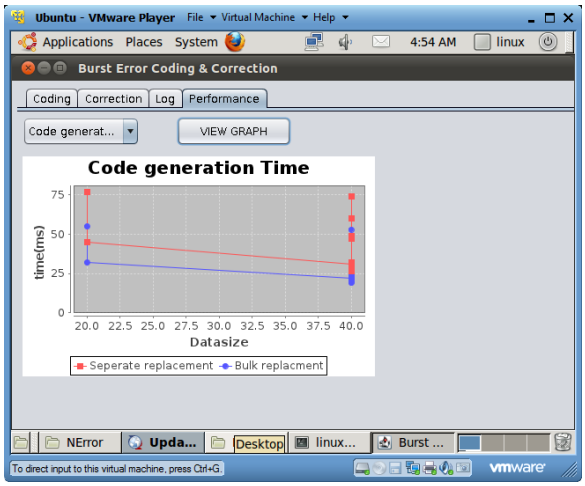

Figure.8 Graph for code generation time

Table 3: Error correction time for different blockcontent file size

S.no	Size of Block File	Existing Error Correction Time(ms)	Proposed Error Correction Time(ms)
1	4	1689	1080
2	8	1617	1128
3	12	1622	1092
4	16	1631	1111
6	20	1598	1108
7	24	1578	1103

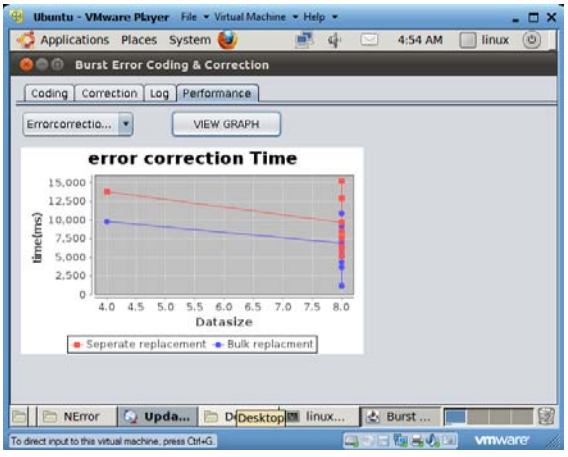

Figure.9 Comparison graph for error rectification time

6. CONCLUSION

In this paper, a rapid and effective spatial-temporal procedure for error identification and rectification is proposed to locate error bursts in big data sets on clouds thereby improving network performance. Error limitation calculation is used to discover location and sources of error in framework diagram. From this point, the identification and rectification technique can be fundamentally stimulated. Moreover, the detection and rectification endeavor can be applied to cloud stage to manage WSNs with a huge number of sensors. Through the examination on cloud computing phase of U-Cloud, it is demonstrated that our proposed technique has proved to be a perfect opportunity for error detection and rectification on a very basic level in bigdata collections made by broad scale sensor framework structures with satisfactory error recognizing accuracy. To justify that our results are efficient, comparison graphs are generated for time efficiency, accuracy and performance Henceforth the recognition and area process can be significantly fastened. Besides, the recognition and area projects can be conveyed to cloud stage to completely exploit the calculation control and big storage. It is exhibited that proposed approach can fundamentally lessen the ideal opportunity for error identification and area in big data indexes produced by expansive scale sensor organize frameworks with adequate acceptable error detecting accuracy, code generation time and error correction time.

The proposed model simulates the error identification and compares to previous existing error identification of network systems in big data. We present a big data error identification and rectification approach by grouping the nodes in blocks, and its integration with Hadoop's Mapreduce. The proposed approach is developing the effectiveness and steady quality of big data on cloud storage. From the outcome of the investigation, it is decided that the proposed method is more appropriate than existing for higher execution. The approach proposed in this paper can significantly reduce the computation and communication time for error detection and correction in big data sets generated by big sensor systems with better accuracy, code generation time and error detection time.

7. REFERENCES

1. S.Tsuchiya, Y.Sakamoto, Y.Tsuchimoto and V.Lee, "Big Data Processing in Cloud Environments," FUJITSU Science and Technology J., Vol. 48, No. 2, April 2012.
2. A. Hadoop. Available: http://hadoop.apache.org/
3. K. Shvachko, K. Hairong, S. Radia and R. Chansler, "The Hadoop Distributed File System," in Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on, 2010.
4. Jeffrey Shafer, Scott Rixner, and Alan L. Cox, "TheHadoop Distributed Filesystem: Balancing Portability and Performance", 2010 IEEE international symposium, March 30th 2010.
5. J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," Commun. ACM, Vol. 51, 2008.
6. C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing, in Proc. 30st IEEE Transactions on Parallel and Distributed Systems, January 2014.
7. Chi Yang, Chang Liu, Xuyun Zhang, Surya Nepal, and Jinjun Chen, "A Time Efficient Approach for

Detecting Errors in Big Sensor Data on Cloud," IEEE Trans. Parallel and Cloud file Systems, Vol. 26, February 2015.

8. S. Mukhopadhyay, D. Panigrahi, and S. Dey, "Model Based Error Correction for Wireless Sensor Networks," IEEE Trans. Mobile Computing, vol. 8, no. 4, pp. 528-543, Sept. 2008.
9. A. Sheth, C. Hartung, and Richard Han, "A Decentralized Fault Diagnosis System for Wireless Sensor Netowrks," Proc. IEEE Second Conf. Mobile Ad-hoc and Sensor Systems (MASS '05), Nov. 2005.
10. Xue Qin, Brian Kelley, Mahdy Saedy, "A Fast Map-Reduce Algorithm for Burst Errors in Big Data Cloud Storage," 2015 10th System of Systems Engineering Conference (SoSE), 2015 IEEE.