ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 10, March 2017

Modified Self Healing Mechanism for Big Data Cloud Architecture

Ms.J.SURIYA [1], UG Scholar ,Department of Information Technology,IFET College of Engineering,Villupuram.
Ms.F.ANISHYA[2] ,M.E, Assistant professor,Department of Information Technology,IFET College of Engineering, Villupuram.
Ms.S.KIRUTHIGA[3], *M.E*, Assistant professor, Department of Information Technology,IFET College of Engineering, Villupuram.

Abstract*— An objective of the paper is to automatically monitor and detect any attacks such as flat line error, out of bound error in the big data network. It also used to resolve the attack on the server. To maintain a definite quality of data in the big data storage atmosphere, self-healing systems can be used. Self-healing systems must complete their tasks in a sensible time. The combination of a big volume of data and the limitation of time requires a big data approach to the problem of self-healing. This commentary analyses the data that self-healing uses as input and justifies its categorization as big data. It is used to troubleshoot the problem and diagnose the problems in the big data network and automatically check the properties and to resolve the problem.

Keywords—Detection; Diagnosis; selfhealing; Troubleshoot

I.INTRODUCTION

Recently, we enter a new period of data explosion which brings about new challenges for big data processing. In general, big data is a collection of data sets so large and complex that it grow into problematic process with on hand database controlling systems or outdated data processing applications. Big data is a collection of data sets so large and complex that it develops difficult to process with database organization systems or traditional data processing applications. It represents the progress of the human cognitive processes, usually includes data sets with dimensions outside the ability of contemporary technology, technique and theory to capture, achieve, and process the data within a tolerable elapsed time. Some work has been done for big data inquiry and error revealing in complex networks including intelligence sensors networks. There are related works in complex network systems data error detection and debugging with online data processing techniques. Self-organizing networking (SON) is a set of principles and concepts to add automation to mobile networks, so that they require less maintenance than traditional networks while improving quality of service (QoS). SON functions take as input the measurements generated by mobile networks, and produce output that depends on the purpose of the function. SON functionalities can be classified in three large categories:

•**Self-configuration**: functionalities that systematize the design and disposition of the network

• **Self-optimization**: functionalities that keep the configuration constraints always working at the ideal level to offer the best

• **Self-healing**: functionalities that automate the solution of complications, reducing anthropological interference and minimizing downtime. Self-healing includes fault discovery, diagnosis, compensation, and recovery. Self-healing aims to automate troubleshooting which is one of the most important tasks. The main objective of troubleshooting is finding and fixing malfunctions in the network. Self-healing f functions include data processing algorithms that analyze data in order to identify and fix problems. Self-healing algorithms are frequently executed using knowledge-based systems (KBSs) that reproduce the process of anthropoid experts in order to complete a task. KBSs are algorithms composed of two parts:

• **Knowledge base** (KB): a codified representation of the field knowledge, that is, the knowledge that the experts need in order to complete the task. To generate and improve the KB, a continuous data mining (DM) process is run in the batch layer

• **Inference engine** (IE): the procedures that use the KB in order to complete the task. The IE conforms to the speed layer. Some KBSs that have been earlier used in troubleshooting are fuzzy logic or Bayesian networks. It is important to follow the procedures of big data when designing these implementations For an algorithm to be parallelizable, its design must pledge that the final outcome is the same when it is run as a single process and when the task is divided among numerous occurrences. Cloud computing is associated with new paradigm for the provision of computing infrastructure and big data processing method for all kinds of resources. Moreover, some different cloud-based expertise has to be accepted because dealing with big data for concurrent processing is difficult.

Big Data Security:

By using operational big data application, a lot of enterprises can significantly reduce their IT cost. However, security and privacy affect the entire big data storage and processing, since there is a massive use of third-party facilities and arrangements that are used to host important data or to perform critical operations. The scale of data and applications grow exponentially, and bring huge experiments of vibrant data monitoring and security protection. Unlike outdated security method, security this complex circumstance. In addition, legal and regulatory issues also need attention in big data is mainly in the form of how to process data drawing out without revealing complex information of users. Besides, current technologies of privacy protection are mainly based on static data set, while data is always vigorously changed, including statistics design, variation of attribute and addition of new data.

II. PROPOSED SYSTEM

The proposed system aims to automatically monitor the big data network and also to resolve any abnormal activities in the network. Our proposed approach, the error detection is based on the scale-free network topology and most of detection

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 10, March 2017

operations can be conducted in limited sequential or spatial data blocks instead of a whole big data set. Hence the detection and location process can be intensely enhanced. Furthermore, the detection and location tasks can be scattered to cloud platform to fully exploit the computation power and massive storage. Through the experiment on our cloud computing raised area of U-Cloud, it is demonstrated that our proposed approach can expressively reduce the time for error detection and location in big data sets produced by large scale sensor network systems with acceptable error detecting accurateness. We aim to develop a different error detection method by exploiting the massive storage, scalability and computation power of cloud to identify faults in big data sets from sensor networks. Fast exposure of data errors in big data with cloud remains challenging. Especially, how to use the computation power of cloud to quickly find and locate errors of nodes in WSN needs to be explored. As a consequence of the decrease in the values of storage hardware, as well as the increase in bandwidth in mobile networks and the growth in the number of connected electronic devices, the volume of data caused by our society is increasing exponentially. All these data encompass information about a wide spectrum of aspects that may be interesting for all types of businesses.

III.PROPOSED SYSTEM ARCHITECTURE

Self healing in big data cloud is based on the following Modules**:**

1. Initialization on cloud
2. Big Data Processing on Cloud
3. Error Definition and Modeling
4. Error Detection &Self Healing

1. Initialization on cloud: Cloud computing infrastructure is becoming popular because it provides an open, flexible, scalable and reconfigurable display place. The proposed error discovery method in this paper will be based on the classification of error types. Specifically, nine types of numerical data abnormalities/errors are registered and familiarized in our cloud error discovery approach. The well-defined error model will generate the error detection process. Compared to previous error detection of sensor network systems, our approach on cloud will be designed and developed by exploiting the enormous data processing capability of cloud to enhance error detection speed and real time reaction. In addition, the structural design feature of compound networks will also be analyzed to combine with the cloud computing with a more efficient way. Based on current research literature review, we divide complex network systems into scale-free type and non scale-free type non scale-free type. Sensor network is a kind of scale-free complex network system which matches cloud scalability feature. Our proposed error detection approach on cloud is specifically clipped for finding errors in big data sets of sensor networks. The main contribution of our proposed detection is to achieve significant time performance improvement in error detection without cooperating error detection accuracy.

2. Big Data Processing On Cloud: Big data has become a fundamental and critical challenge for modern society. Cloud computing provides an ideal platform for big data storage, dissemination and interpreting with its massive computation power. Map Reduce has been widely revised from a batch processing framework into a more incremental one to examine huge-volume of incremental data on cloud. It is a framework for processing parallelizable problems across big data sets using a large number of computers (nodes), collectively referred to as a collection in which all computers (nodes) are on the same local network and use similar hardware; or a grid in which the nodes are shared across geographically and administratively distributed systems.

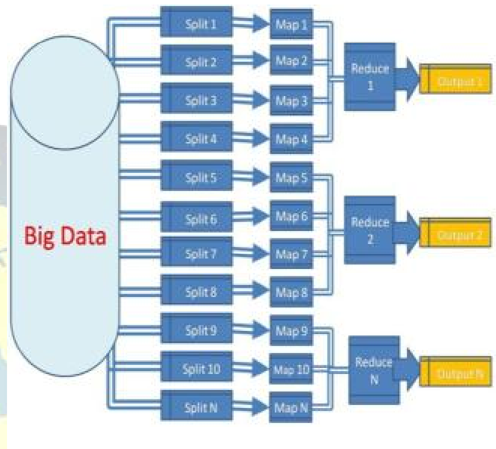

Fig1 Big data processing

It can sort a petabyte of data in only rare hours. The parallelism also offers some opportunity of improving from partial failure of servers or storage during the operation.

3. Error Definition and Modeling:

Considering the specific feature of numeric data errors, there are several abnormal data scenarios .The "flat line faults" indicates a time series of a node in a network system keeps unchanged for unacceptable long time duration. In real world applications, sampled data and transmitted data always have slight changes with the time flow. The "out of data bounds faults" indicates impossible data values are observed based on some sphere of influence knowledge. In real world applications, if a temperature value of water is reported C, it can be treated as a data fault directly. The "data lost fault" means there are missing data values in a time series during the data generation or communication. The time series with "data lost fault" normally needs data cleaning, the "spike faults" indicates in a time series data items which are totally out of the prediction and normal changing trend. Because the above four types of errors can happen both at data generation and exchange stages, the error types can also be categorized into node side and edge side separately. Combining the data faults scenarios in we present a classification of complex network systems data errors based on the time series analysis available in the online supplemental material where n is ID of node in network systems. T represents windowpane extent of time series. f(n, t) is numerical values collected within window t from node n. g(n, l) is allocation function which records the

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 10, March 2017

cluster, the data source node and partition situation related to node n. We present classification, definition of error type to guide error detection algorithm. Suppose a data record from node is denoted as r(p,q, f(n, t), g(n, l)), where n is ID of node in network systems. T represents window length of time series. f(n, t) is numerical values collected within window t from node n. g(n, l) is allocation function which records the cluster, the data source node and partition situation related to node n.

Definition1 (Node/Edge side flat line error).

Let ri(pi,qi),f(pi,qi), g(ni,l) be a time series recorded from, where I is a time stamp. If any element x ≡ δi, where δi is an node ni effective constant during time window t, x € f(pi,qi) and g(ni,l)=0/g(ni,l) !=0, ni is the data source node, there is an node side/edge side flat line error. g(ni,l)=0/g(ni,l) !=0, ni is the data source node, there is an node side/edge side flat line error.

Definition 2 (Node/Edge side data lost error).

Let ri (pi,qi) ,f(pi,qi), g(ni,l) be a time series recorded from node ni, where I is a time stamp.If f(pi,qi) = null && t i > ι ,ι is the time duration from outside application requirement, and if g(ni,l) =0/ g(ni,l) != 0, ni is the data source node, the error is a node/edge side data lost error.

Definition 3 (Node/Edge side out of bounds error).

Let ri (pi,qi), f(pi,qi), g(n, l) be a time series record from node ni, where i is a time stamp. If any element x > θ , x € f(pi,qi), θ is a threshold defined from the application requirement, and if g(ni,l) = 0/ g(ni,l) != 0; ni is the data source node, the error is a node/edge side out of bound error.

Definition 4 (Node/Edge side spike error).

Let ri (pi,qi),f(pi,qi), g(n, l) be a time series record from node ni, where i is a time stamp. If | f(pi,qi) – fp(pi,qi)| / ti > Ψ ,Ψ is the acceptable changing trend, fp(pi,qi) is the predicted time series with an adopted prediction model, and if g(ni,l) = 0/ g(ni,l) != 0, ni is the data source node, the error is a node/edge side spike error.

4. **Error Detection and Localization:**

We propose a two-phase approach to conduct the computation required in the whole process of error detection and localization.

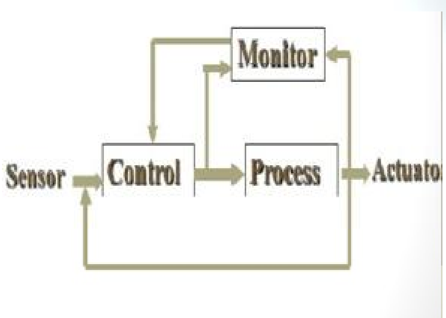

Fig .2 Monitoring The Process

At the phase of error detection, there are three inputs for the error detection algorithm. The first is the graph of network. The second is the total collected data set D and the third is the defined error patterns p. The output of the error detection algorithm is the error set D'. After the error pattern matching and error detection, it is important to locate the position and source of the detected error in the original WSN graph G(V, E). The input of the Algorithm 2 is the original graph of a scale-free network G(V, E), and an error data D from Algorithm 1. The output of the algorithm 2 is G'(V', E') which is the subset of the G to indicate the error location and source. Self-healing implementations work by detecting disruptions, diagnosing failure root cause and deriving a remedy, and recovering with a sound strategy'. However, like in many other publications, also in the term repair/self-repair is used without definition, and sometimes in a confusing way.

Fig.3 self healing mechanism

In this modified self healing mechanism activating monitor is used for monitor each server in the network.

CONCLUSION

In order to detect errors in big data sets from sensor network systems, a novel approach is developed with cloud computing. Firstly error classification for big data sets is presented. Secondly, the correlation between sensor network systems and the scale-free complex networks are introduced. According to each error type and the features from scale-free networks, we have proposed a time-efficient strategy for detecting and locating errors in big data sets on cloud. With the experiment results from our cloud computing environment U-Cloud, it is demonstrated that,

1) The proposed scale-free error detecting approach can significantly reduce the time for fast error detection in numeric big data sets

2) The proposed approach achieves similar error selection ratio to non-scale-free error detection approaches. In future, in accordance with error detection for big data sets from sensor network systems on cloud, the issues such as error correction, big data cleaning and recovery will be further explored.

References

[1] *NGMN, Use Cases Related to Self-Organizing Network, Overall Description, Apr. 2007.*

[2] 3GPP, "Telecommunication Management; Self-Organizing Networks (SON); Self-Healing Concepts and Requirements,"Rel. 11, TS 32.541.

[3] R. Barco, P. Lázaro, and P. Munoz, "A Unified Framework for Self-Healing in Wireless Networks," IEEE Commun. Mag., vol. 50, no. 12, Dec. 2012,pp. 134–42.

[4] E. J. Khatib et al., "Data Mining for Fuzzy Diagnosis Systems in LTE Networks,"Expert Systems with Applications, vol. 42, no. 21, 2015, pp. 7549–59.

[5] P. Szilagyi and S. Novaczki, "An Automatic Detection and Diagnosis Framework for Mobile Communication Systems," IEEE Trans. Network and Service Management, vol. 9, no. 2, June 2012, pp. 184–97 [6] N. Baldo, L. Giupponi, and J. Mangues-Bafalluy, "Big Data Empowered Self Organized Networks," Proc. Euro. Wireless 2014, VDE, 2014, pp. 1–8.

[7] A. Imran and A. Zoha, "Challenges in 5G: How to Empower SON with Big Data for Enabling 5G," IEEE Network, vol. 28, no. 6, Nov. 2014, pp. 27–33.

[8] P. Russom, "Big Data Analytics," TDWI Best Practices Report, 4th qtr. 2011.

[9] N. Marz and J. Warren, Big Data: Principles and Best Practices of Scalable Realtime Data Systems, 1st ed., Manning, 2015.

[10] J. Dean and S. Ghemawat, "Mapreduce: Simplified Data Processing on Large Clusters," Commun. ACM, vol. 51, no. 1, Jan. 2008, pp. 107–13.

[11] R. Cattell, "Scalable SQL and NoSQL Data Stores," ACM SIGMOD Record,vol. 39, no. 4, 2011, pp. 12–27.

[12] R. Barco et al., "Continuous Versus Discrete Model in Autodiagnos is Systems for Wireless Networks," IEEE Trans. Mobile Computing, vol. 7, no. 6, June 2008, pp. 673–81.

[13] E. J. Khatib et al., "LTE Performance Data Reduction for KnowledgeAcquisition," Proc. GLOBECOM Wksps, Dec. 2014, pp. 270–74.

[14] I. de la Bandera et al., "Cell Outage Detection Based on Handover Statistics,"IEEE Commun. Lett, vol. 19, no. 7, 2015, pp. 1189–92.

[15] P. Muñoz et al., "Fault Diagnosis in Networks," Patent424,2015. C. Grosan and A. Abraham, "Rule-based expert systems," in Intelligent Systems, ser. Intelligent Systems Reference Library, Berlin, Germany:Springer-Verlag, 2011, vol. 17, pp. 149–185.[16] T. Ross, Fuzzy Logic With Engineering Applications. NewYork,NY,
USA: Wiley, 2010.

[17] M. Makrehchi, O. Basir, and M. Kamel, "Generation of fuzzy membership function using information theory measures and genetic algorithm,"
in Proc. Int. Fuzzy Syst. Assoc. World Congr. Conf. Fuzzy Sets Syst., 2003,
pp. 603–610.

[18] I. Derbel, N. Hachani, and H. Ounelli, "Membership functions generation based on density function," in Proc. IEEE Int. Conf. Comput. Intell.
Security, 2008, vol. 1, pp. 96–101R. Barco, V. Wille, L. Díez, and M. [19]Toril, "Learning of model parameters
for fault diagnosis in wireless networks," Wireless Netw., vol. 16, no. 1,pp. 255–271, 2010.

[20] J. A. Rice, Mathematical Statistics and Data Analysis. Pacific Grove,CA,USA:Brooks/Cole,1998. [i] S. Tsuchiya, Y. Sakamoto, Y. Tsuchimoto, and V. Lee,
[21]Big Data Processing in Cloud Environments," FUJITSU Science and Technology J., vol. 48, no. 2, pp. 159-168, 2012.

[22] "Big Data: Science in the Petabyte Era: Community Cleverness Required," Nature, vol. 455, no. 7209, p. 1, 2008.

[23] X. Zhang, C. Liu, S. Nepal, and J. Chen, "An Efficient Quasi-Identifier Index Based Approach for Privacy Preservation over Incremental Data Sets on Cloud," J. Computer and System Sciences,vol. 79, pp. 542-555, 2013.

[24] W. Dou, X. Zhang, J. Liu, and J. Chen, "HireSomeTowards Privacy-Aware Cross-Cloud Service Composition for Big Data Applications," IEEE Trans. Parallel and Distributed Systems, 2013