Using Non-Parametric Method for Detecting Performance Anomaly in Large Scale Computing

P.Aravind[1], Ms. A. Dhivya Bharathi[2]

[1, 2] Department of Information Technology

[1, 2] M.A.M College of Engineering, Trichy, India.

Abstract- *The proposed system anomaly detection on node by using scalability, non-parametric method involves large scale computing. It follow some method to detect anomaly node .First method is used to decentralized process based on hierarchical grouping based on divide and conquer over the network. Then second method is hardware homogeneity i.e., (group1, group2…... group n).Simultaneously some feature extraction added to the group based on local communication and executed with non-parametric clustering of parallel analysis. Final method is two-phase majority voting i.e., the node label with "M" _label of all node. Node label with "N" i.e., normal node but node label with "A" i.e., abnormal node.*

Keywords- Anomaly detects Divide and Conquer, Homogeneity, Two-phase Majority.

I. INTRODUCTION

Anomaly detection is used to identify the hardware and software fault, application bugs etc. It is more difficult to remove fault node. The proposed system using group of strategy for divided big problem involved to analyze large system into many small problems using divide and conquer model. Then next non-parametric clustering method to implement for fault identification. The number of nodes in each group and local Communication using decentralized design of high scalability. Finally, two phase majority voting mechanism used to improve anomaly detection. In each group, we collect data to characterize node behaviors, and transfer them into a uniform format for further analysis. The data gathered from each group are put into a m _ n matrix X, where m is the number of features (rows) per node and n is the number of nodes (columns) in the group. The value of m can be further represented as m ¼ c _ t, where c is the number of features gathered to characterize node behaviors and t is the number of snapshots sampled per node. As the collected data have different scales, the matrix X is normalized across columns such that all feature values fall into the range of 0.0 and 1.0. A scalable, non-parametric method for effectively detecting performance anomalies in large-scale systems. The design is generic for anomaly detection in a variety of parallel and distributed systems exhibiting peer-comparable property. It adopts a divide-and-conquer approach to address the scalability challenge and explores the use of non-parametric clustering and two-phase majority voting to improve detection flexibility and accuracy. The derive probabilistic models to quantitatively evaluate our decentralized design. The detection is based on both OS level (black-box) performance metrics and middleware level (white-box) performance metrics. In addition, an important feature of this work is application-transparent, meaning that our method does not require any modifications of the hosted applications. Distributed computing like Map/Reduce may also form a peer-comparable environment in case of homogeneous hardware resources.

II. EXPERIMENTAL PROCEDURE

1. Decentralized process:

As the number of nodes in each group is limited and only local communication is needed for problem solving, such a decentralized design is able to achieve high scalability. This process is more useful to hierarchical group of each node. First, adopt a grouping strategy, through which we divide the big problem involving the analysis of a large system into many small problems while at the same time maintain the peer comparable environment.

Second, we explore a non-parametric clustering method that does not rely on any pre-defined cluster numbers and thus is capable of handling multiple anomalies.

1) Divide and Conquer method:

This method is used to split large problem into small problem based on divide and conquer method. Then basic concept of hierarchical grouping is used to split and merge node follow on three types.

2) Geographical grouping:

Computing nodes are grouped using geographical locations.

3) Topology-aware grouping:

Computing nodes are further divided based on their network topologies and hardware configurations.

4) Random:

Finally, every node serves as a central node and forms a group by randomly assigning n neighbors to it.

2. Advantages:

Two-phase majority voting mechanism is more useful for identify fault node and also remove easily. Performance anomalies caused by application bugs, hardware or software faults, or resource contention can have great impact on system-wide performance and could lead to significant economic losses for service providers. Feature extraction is used. In each group, we collect data to characterize node behaviors, and transfer them into uniform format for further analysis.

- Precisely rate servers cost efficiency with respect to load characteristics.
- Average cost, including the expense for purchasing server and electricity.
- Low cost
- System failure recovers easily. High performances.
- Communication of normal node easily identified.
- More easily to implement divide and conquer method

III. RESULTS AND DISCUSSIONS

1. Hierarchical group:

The purpose of hierarchical grouping is to avoid global computation and communication for decision making, and to guarantee a peer-comparable environment within each group. Geographical grouping is conducted first, which is applied to avoid the long distance communication between remote nodes. The goal of topology-aware grouping is to further reduce the group size and maintain hardware homogeneity. The rule in this step varies according to different system environments.

2. Feature Extraction:

In each group, we collect data to characterize node behaviors, and transfer them into a uniform format for further analysis. The data gathered from each group are put into a m _ n matrix X, where m is the number of features (rows) per node and n is the number of nodes (columns) in the group. The value of m can be further represented as m ¼ c _ t, where c is the number of features gathered to characterize node behaviors and t is the number of snapshots sampled per node. As the collected data have different scales, the matrix X is normalized across columns such that all feature values fall into the range of 0.0 and 1.0.

3. Non-Parametric Clustering:

Clustering analysis is used to distinguish node behaviors within the same group. Commonly adopted clustering methods can be categorized into four types, including centroid-based (e.g., k-means), connectivity-based (e.g., hierarchical), distribution-based (e.g., Gaussian mixture) and density-based (e.g., DBSCAN and mean-shift).

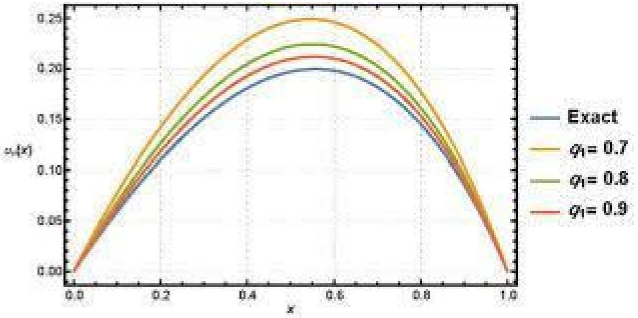

Figure 1.

4. Two-Phase Majority Voting:

Based on the clustering, the next component of our design is two-phase majority voting, aiming to identify abnormal nodes in each group. In the first phase, a node is labeled with (“Majority”) if it belongs to the majority of all group

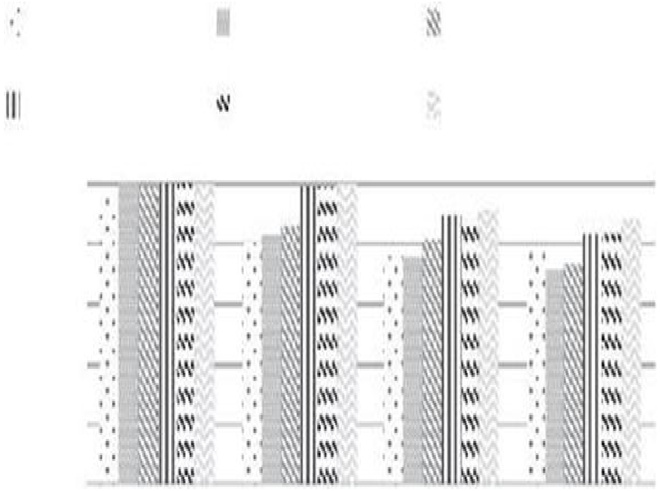

Figure 2.

members; otherwise it is labeled with F(“Fewness”). In the second phase, only the nodes labeled with M have the right to vote.

A node is labeled with N (“Normal”) if it belongs to the majority of the group members labeled with M; otherwise, it is labeled with A (“Abnormal”).

IV. CONCLUSION

We have presented a practical and scalable anomaly detection method for large-scale systems. Our design features a decentralized approach based on hierarchical grouping, non-parametric clustering, and two-phase majority voting. Experimental results have demonstrated that the proposed method can provide high detection accuracy by effectively distinguishing distinct anomaly patterns, with a negligible overhead. The proposed design is applicable to a variety of parallel and distributed computing environments with the peer-comparable property. Our ongoing work includes evaluating the proposed detection method in other large scale computing environments. The number of nodes in each group and local communication using decentralized design of high scalability. Finally, two phase majority voting mechanism used to improve anomaly detection. In each group, we collect data to characterize node behaviors, and transfer them into a uniform format for further analysis. The data gathered from each group are put into a m _ n matrix X, where m is the number of features (rows) per node and n is the number of nodes (columns) in the group. The value of m can be further represented as m ¼ c _ t, where c is the number of features gathered to characterize node behaviors and t is the number of snapshots sampled per node.

First method is used to decentralize process based on hierarchical grouping based on divide and conquer over the network. Then second method is hardware homogeneity. Final method is two-phase majority Voting.

REFERENCES

[1] Hadi M. Investigation on turning of AISIH13 with applying minimum quantity lubricant, Indian Journal of Science and Technology, 2013; 6 (2); 4094-7

[2] Obikawa T Machining with least quantity lubrication, Advanced Machining Technologies, 2014; 11:255-281

[3] Liao YS, Lin HM and Chen YC, Feasibility study of the minimum steely by coated carbide tool, International Journal of Machine Tools and Manufacture, 2007; 47-1667-76

[4] Nam JS, Lee P-H and Lee SW, Experimental characterization of micro-drilling process using nanofluids minimum quantity lubrication, International Journal of Machine Tools and Manufacture, 2011; 51;649-52

[5] Rahmati B, Sarhan A.A and Sayuti M (2014), Investing the optimum MoS2 nano lubrication parameters in CNC milling of AL6061-T6 alloy, The International Journal of Advanced Manufacturing Technology.

[6] Vasmi Krishna P, Srikant R.R and Nageswara Rao D (2010), Experimental investigation on the performance of nanoboric and suspensions in SAE-40 and coconut oil during turning of AISI 1040 steel, IJOMTM, 50 (10), 911-916

[7] Prasad M.M.S and Srikant R.R, Performance evalution of nanographite inclusions in cutting fluids with MQL technique in turning of AISI 1040 steel, 381-393

[8] Vasu V, Reddy GPK. Effect of MQL with Al2O3 nano particles on surface roughness, tool wear and temperature dissipations in machining Inconel 600 alloy, Journal of Nano Engineering and Nano System, 2011; 225 (1): 3-16.

[9] Lee P.H, Nam J.S, Li C and Lee S.W, An experimental study on micro-grinding process with nanofluid minimum quantity lubrication, International Journal of Precision Engineering and Manufacture, 2012; 13 (3); 331-338

[10] Saravana Kumar N, Prabu L and Karthik M, Experimental analysis on cutting fluid dispersed with silver nano particles, Journal of Mechanical Science and Technology 2014; 28 (2); 645-651

[11] Sayuti M, Sarhan A.A and Salem F, Novel uses of SiO2 nano lubrication system in hard turning process of hardened steel AISI 4140 for less tool wear, surface roughness and oil consumption, Journal of Cleaner

[12] Mostafa Hadi and Reza Atefi, Effect of MQL lubrication with Gamma-Al2O3 nano particles on surface roughness in Milling AISI D3 steel, Indian Journal of Science and Technology, Vol 8 (3), 296-300, 2015.

[13] W. M. Rand, “Objective criteria for the evaluation of clustering methods,” J. Am. Statist. Assoc., vol. 66, pp. 846–850, 1971. Production, 2014; 67; 265-276 S. Ramaswamy, R. Rastogi, and

[14] Shim, “Efficient algorithms for mining outliers from large data sets,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2000, pp. 427–438.

[15] M. M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander, “LOF: Identifying density-based local outliers,” in Proc. ACM SIGMOD Int.1.Conf. Manage. Data, 2000, pp. 93–104.