A Scalable, Non-Parametric Anomaly Detection Framework for Hadoop

Li Yu and Zhiling Lan
Department of Computer Science
Illinois Institute of Technology
Chicago, IL 60616
{lyu17, lan}@iit.edu

ABSTRACT

In this paper, we present a scalable and practical problem diagnosis framework for Hadoop environments. Our design features a decentralized approach based on hierarchical grouping and a novel non-parametric diagnostic mechanism. We evaluate our framework under various Hadoop workloads. The experimental results show that our design outperforms traditional methods significantly in the context of complex anomaly patterns and high anomaly probability.

Categories and Subject Descriptors

D.2.5 [**Software Engineering**]: Testing and Debugging—*Diagnostics*

General Terms

Reliability

Keywords

Hadoop, Problem Diagnosis, Hierarchical Grouping, Non-parametric Clustering

1. INTRODUCTION

Cloud computing becomes a popular computing paradigm and starts to make significant impact on various domains. The MapReduce framework along with its open source implementation on Hadoop has been widely used by many businesses such as Amazon and Yahoo for large-scale computing [1]. Despite the flexibility and convenience offered by such an environment, performance anomalies such as *resource contentions, application bugs, and hardware/software failures* are commonly observed by Hadoop users. These performance problems cause some nodes taking longer time to finish their assigned tasks, thereby resulting in financial penalty to both companies and users. Hence, an effective problem diagnosis mechanism is essential for Hadoop environments.

Currently anomaly detection in large-scale systems faces two major challenges. First, existing anomaly detection methods typically rely on a centralized design that uses a central manager for data collecting data and decision making. Although these schemes provide good diagnosis accuracy, they fail to meet the scalability requirement. Second, existing schemes tend to use parametric methods for anomaly detection. Parametric methods assume the number of behavior patterns is known in priori [4]. Unfortunately, this requirement is hardly met in practice. In fact, problematic nodes may behave differently due to distinct root causes.

In this paper, we present a proof-of-concept study of a novel framework for anomaly detection. It explores non-parametric clustering and majority based voting to address the aforementioned challenges.

2. METHODOLOGY

Our design consists of three main components, namely hierarchical grouping, non-parametric clustering and two-phase majority voting. Hierarchical grouping is built on the well-known divide-and-conquer philosophy, and it enables our framework to minimize global communication for decision making. Non-parametric clustering and two-phase majority voting do not require any predefined number of clusters.

The first step of our method is hierarchical grouping, which is built on the well-known divide-and-conquer philosophy, and it enables our framework to minimize global communication for decision making. At the top level, computing nodes are grouped according to their geographical locations if tasks are allocated remotely. Next at the lower level, computing nodes within the same location are divided based on their network connections. The grouping rules at this level vary according to different network topologies. If a group at this level still contains a large number of nodes, a random selection strategy is used to further partition the group. For each node, we use it as a central point and form a group by randomly assigning n neighbors to it. Using this strategy, a node may belong to one or more groups, but its status is only determined by the group where it is the central node.

After the hierarchical grouping, our design will perform group analysis (i.e., non-parametric clustering and two-phase majority voting) concurrently. For each group, we first collect system data characterizing node behaviors and transfer them into a uniform format for further analysis. Next we project the high-dimensional data to a lower feature space using kernel principal component analysis (KPCA) [5].

Then for the nodes in each group, clustering is used to assign them into "clusters" so that the nodes within the same cluster are more similar to each other than to those in other clusters. Rather than using the well-known parametric clustering methods such as k-means or hierarchical clustering [2], we explore a non-parametric clustering method called *Adaptive Mean Shift Clustering*, which does not require predefined number of clusters in advance [3].

Based on the clustering, we propose a two-phase major-

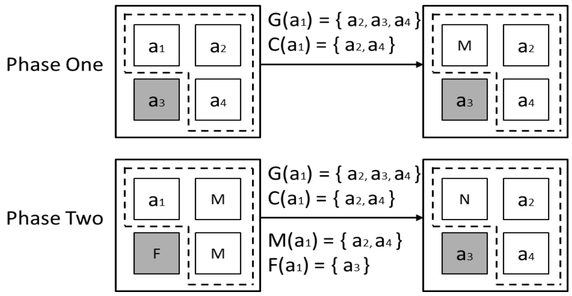


Figure 1: In this example, there are two clusters $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_4\}$ and $\{\mathbf{a}_3\}$ in the group, and $\mathbf{a}_1$ is the central node. In phase one, $\mathbf{a}_1$ is labeled with M. In phase two, we assume $\mathbf{a}_2$ and $\mathbf{a}_4$ are labeled with M and $\mathbf{a}_3$ is labeled with F during the first phase, then $\mathbf{a}_1$ is finally labeled with N.

ity voting mechansim, aiming to identify abnormal nodes in each group. In phase one, a node is labeled with M ("Majority") if it belongs to the majority of all group members; otherwise it is labeled with F ("Fewness"). In phase two, a node is finally labeled with N ("Normal") if it belongs to the majority of group members labeled with M in the first phase; otherwise, it is labeled with A ("Abnormal"). Only the nodes labeled with M in the first phase have the right to vote in the second phase. Figure 1 gives an example of the two-phase majority voting process for a group, in which $\mathbf{a}_1$ is the central node.

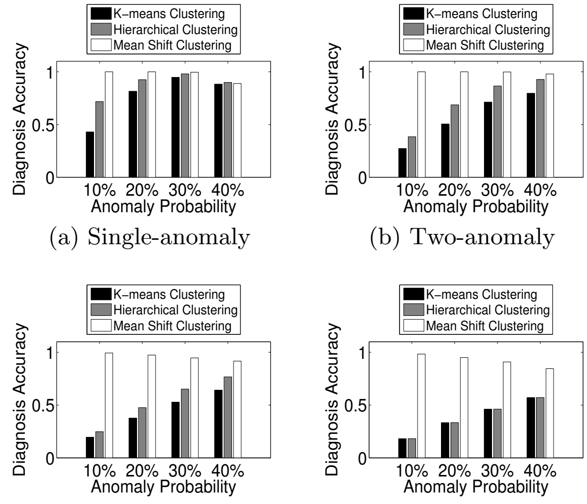
3. EXPERIMENTS

The experiments were conducted on a 65-node cluster located at Illinois Institute of Technology. It consists of 64 computing nodes and one head node. Each computing node has two Quad-Core AMD Opteron(tm) processors, 8GB memory and a 250GB 7200RPM SATA-II disk. All nodes are equipped with Gigabit Ethernet interconnection.

We used three Hadoop (version 0.20.2) workloads, namely *WordCount* and *TeraSort* and *Pig* in our experiments and injected four anomaly patterns: three caused by resource consumption at the system level and one caused by bugs from the application level. The individual anomaly patterns (single-anomaly) and their combinations (multiple-anomaly, from 2 to 4) are injected randomly into the testbed with ten anomaly probabilities $(5\%, 10\%, 15\%, ..., 50\%)$. The injected anomalies include *CPU Hog*, *Net Hog*, *Disk Hog* and *Task Hang*.

Figure 2 presents diagnosis accuracy achieved by different clustering methods, including our non-parametric method, k-means method, and hierarchical method. Our non-parametric clustering method outperforms parametric clustering methods significantly in terms of of diagnosis accuracy given multiple anomaly patterns. The average diagnosis accuracy improvement is 34%.

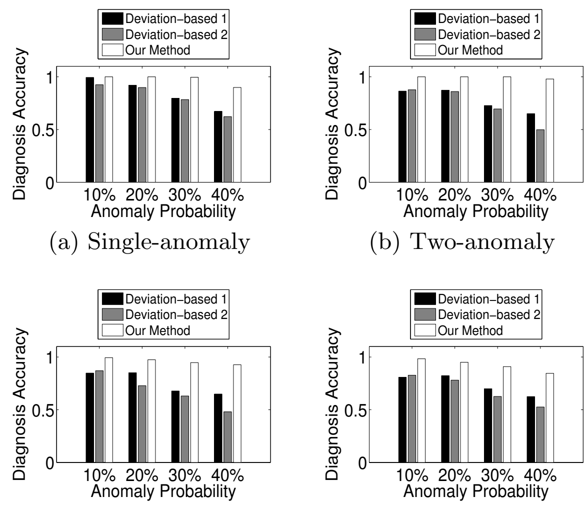
Figure 3 gives the comparison of our method as against other non-clustering based methods. Compared to deviation-based methods, our method has an average of more than 15% improvement in terms of diagnosis accuracy, given multiple anomaly patterns (i.e., other than single-anomaly) and high anomaly probability (i.e., $p > 20\%$).



(a) Single-anomaly (b) Two-anomaly

(c) Three-anomaly (d) Four-anomaly

Figure 2: Comparison of clustering methods in terms of diagnosis accuracy ($F\_measure$).



(a) Single-anomaly (b) Two-anomaly

(c) Three-anomaly (d) Four-anomaly

Figure 3: Comparison of our method and non-clustering based methods in terms of diagnosis accuracy ($F\_measure$).

4. REFERENCES

[1] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. In *Proc. of OSDI*, 2004.

[2] B. S. Everitt, S. Landau, M. Leese, and D. Stahl. *Cluster Analysis*. Wiley, 2011.

[3] B. Georgescu, I. Shimshoni, and P. Meer. Mean shift based clustering in high dimensions: A texture classification example. In *Proc. of ICCV*, page 456, 2003.

[4] I. Laguna, T. Gamblin, B. R. de Supinski, S. Bagchi, G. Bronevetsky, D. H. Anh, M. Schulz, and B. Rountree. Large scale debugging of parallel tasks with automaded. In *Proc. of SC*, 2011.

[5] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.