1. INTRODUCTION
As supercomputers continue to grow in size and complexity, reliability becomes a major concern in the field of high performance computing. Because of this, modern systems are deployed with various monitoring and logging facilities to track system health and status during operations. For example, the environmental monitors deployed on IBM Blue Gene systems can collect data like temperatures, clock frequency, fan speeds, and voltages, from the underlying hardware devices; the OVIS monitoring tool developed from Sandia National Lab can collect various state variables (e.g., temperature, CPU utilization, fan speed) and user-specified variables (e.g., aggregated memory errors over the life span of a job) on various large-scale clusters. Log data is a critical asset to successfully operating large-scale systems. System administrators typically examine these data to spot faults or errors, and numerous technologies are presented to utilize log data for fault prediction and root cause analysis.

2. BACKGROUND
System monitoring is a broad category. There are solutions that monitor for the proper operation of servers, network gear, and applications, and there are solutions that track the performance of those systems and devices, providing trending and analysis. Some tools will sound alarms and notifications when problems are detected, while others will even trigger actions to run when alarms sound. Here we list a collection of open source solutions that aim to provide some or all of these capabilities. Cacti is a very extensive performance graphing and trending tool that can be used to track just about any monitored metric that can be plotted on a graph. From disk utilization to fan speeds in a power supply, if it can be monitored, Cacti can track it and make that data quickly available. Nagios is the old guard of system and network monitoring. It is fast, reliable, and extremely customizable. Nagios can be a challenge for newcomers, but the rather complex configuration is also its strength, as it can be adapted to just about any monitoring task. What it may lack in looks it makes up for in power and reliability. Other tools include NeDi, Observium, Zabbix, Ntop, and etc.

3. METHODOLOGY
In this study, we intend to improve the data analysis capacities of OVIS by enhancing the data quality it collected. Specifically, we will propose a 2-dimensional online data filtering mechanism to remove noisy and redundant data horizontally (via feature selection) as well as vertically (via instance selection). Our design comprises three major components. First is data integration, where various data streams from different sources are synchronized and fed to a buffer in memory. Once the buffer is filled, the data is moved out for further operations and the buffer is vacated to receive new data. Second is feature selection, where data filtering is conducted horizontally, meaning representative features are selected and the rest is dropped. Third is instance selection, where data filtering is performed vertically along the time axis, meaning representative instances are selected and the rest is dropped. To truly make this filtering mechanism useful in realistic environments, our design intends to provide sever-
al key features. First, our design provides online filtering of log data, inspired by the idea of stream processing. Second, the design is general and flexible, where a variety of selection techniques can be integrated in this framework. Third, the design can be used with data compression technologies for further data reduction.

4. EXPERIMENTS

To demonstrate the effectiveness of our filtering design, we conduct case studies on two real environmental data collected from the production supercomputers (an environmental log from a Blue Gene/P system at Oak Ridge National Lab and an OVIS log from a cluster at Sandia National Lab). We examine the amount of disk storage that can be reduced by applying our online filtering mechanism. We also compare the effects of our filtering mechanism as against random filtering. Further, we study whether the proposed filtering mechanism rings a positive or negative impact on failure prediction and diagnosis. The use of multiple data is to ensure the presented mechanism is not biased to any specific system or log and thus is general for providing filtering service for a variety of log data.

For the Blue Gene/P environmental data, our method can reduce 85.6% disk space without losing prediction accuracy and root cause information. For the OVIS log, we can achieve 99.7% disk space saving without losing both prediction accuracy and root cause information. To the best of our knowledge, we are among the first to explore 2-dimensional filtering (i.e., horizontally as well as vertically) for reducing system logs like environmental data and utilize the filtered data for better failure prediction and diagnosis on large-scale systems.

5. CONCLUSION

The data filtering framework proposed in this study is expected to be applicable to different large-scale systems, particularly on numeric system data. Future work can be the extension of this framework to accept other data types. This research report is a summary of our previous work [Yu et al. 2015b; Yu and Lan 2015; Zheng et al. 2015; Yu et al. 2012; Berrocal et al. 2014; Yu et al. 2014; Yu and Lan 2013; Zheng et al. 2012; Zanibbi and Yu 2011; Zheng et al. 2011; Yu et al. 2011; Yu and Zanibbi 2009; Yu 2013; 2012; Yu et al. 2011; Yu et al. 2015a; 2015c; Yu 2015; 2014; 2010] [Zhou et al. 2015; Yang et al. 2013; Yang et al. 2014; Zhao et al. 2015; Zhou et al. ; 2014; Zhou et al. 2014; Berrocal et al. 2015].

REFERENCES


Li Yu, Zhiling Lan, and James M. Brandt. 2011. Enhance Data Diagnosis and Prognosis Capabilities in OVIS (Research Poster). In *IEEE/IFIP International Conference on Dependable Systems and Networks*.

Li Yu, Dong Li, Sparsh Mittal, and Jeffrey S. Vetter. 2014. Quantitatively Modeling Application Resilience with the Data Vulnerability Factor. In *Proceedings of SC*.


---