Real-Time Assessment of Supercomputer Status by a Comprehensive Informative Metric through Streaming Processing

Yawei Hui*, Rizwan A. Ashraf†, Byung H. Park‡, Christian Engelmann§
Computer Science and Mathematics Division, Oak Ridge National Laboratory
{*huiy, †ashrafra, ‡parkbh, §engelmannc}@ornl.gov

***Abstract*—Supercomputers are complex systems used to simulate, understand and solve real-world problems. In order to operate these systems efficiently and for the purpose of their maintainability, an accurate, concise, and timely determination of system status is crucial for its users and operators. However, this determination is challenging due to intricately connected heterogeneous software and hardware components, and due to sheer scale of such machines. In this poster, we demonstrate work-in-progress towards realization of a real-time monitoring framework for the 18,688-node Titan supercomputer at Oak Ridge Leadership Computing Facility (OLCF). Toward this end, we discuss the use of metrics which present a one-dimensional view of the system generating various types of information from 1000s of components and utilization statistics from 100s of user applications in near real-time. We demonstrate the efficacy of these metrics to understand and visualize raw log data generated by the system which otherwise may compose of 1000s of dimensions. We also demonstrate the architecture of proposed real-time stream processing framework which integrates, processes, analyzes, visualizes and stores system log data from an array of system components.**

***Index Terms*—System Monitoring, Quantification Metrics, Reliability, Availability, Serviceability**

I. Introduction

Real-time analysis of system status requires efficient processing of streams of data which is often sporadic but bursty, and is composed of heterogeneous types. With exascale systems on the horizon, the complexity and the number of components and subsystems will continue to grow, generating huge amounts of log data. In order to meet many challenges imposed by volume, diversity and complexity of log data accumulated at OLCF, we have developed a big data analytics platform – Log processing by Spark and Cassandra-based ANalytics (LogSCAN) [1] – that provides flexible and scalable data acquisition, transformation and computation on supercomputer log data. One of the main goals of LogSCAN is to provide simple but effective metrics that represent the status of the system for users and operators near real-time [2]. Previous work demonstrates the benefit of a monitoring framework to improve application performance [3]. Machine learning has also been utilized to diagnose performance variation in applications commonly observed in large-scale computing machines using resource utilization information and data from performance counters [4]. As an extension to our prior work [1], we demonstrate the use of LogSCAN for real-time system monitoring in this work by processing near real time data streams of various RAS events most of which are recorded on all nodes in the system and placement information available for executing applications.

This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

II. System Status Metrics in LogSCAN

Various events parsed by LogSCAN could potentially be used to monitor the system status as they provide the original information. However, it's quite a challenge to inspect the system status by simple occurrence counts of specific event types. Figure 1 shows total counts of 22 event types processed by LogSCAN's parsers. Although the plot may deliver some insight regarding the current system, it cannot represent system status of future systems. What if the dimension of the feature space (or in terms of event types) increases by ten-fold in the future?

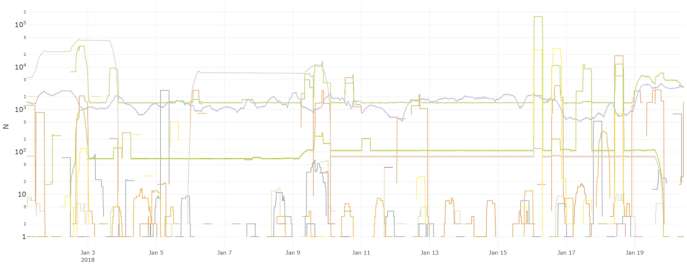

Fig. 1: Event counts for different event types (in different colors) during [Jan. 01, 2018 to Jan. 20, 2018].

It is one of the design goals of LogSCAN to find temporally quantifiable low-dimensional metrics in forms of time series in order to describe the evolutionary history of the system. We design and implement two metrics to adequately capture the state of the supercomputer at any given instant: 1) *System Information Entropy (SIE)* metric, which captures the state

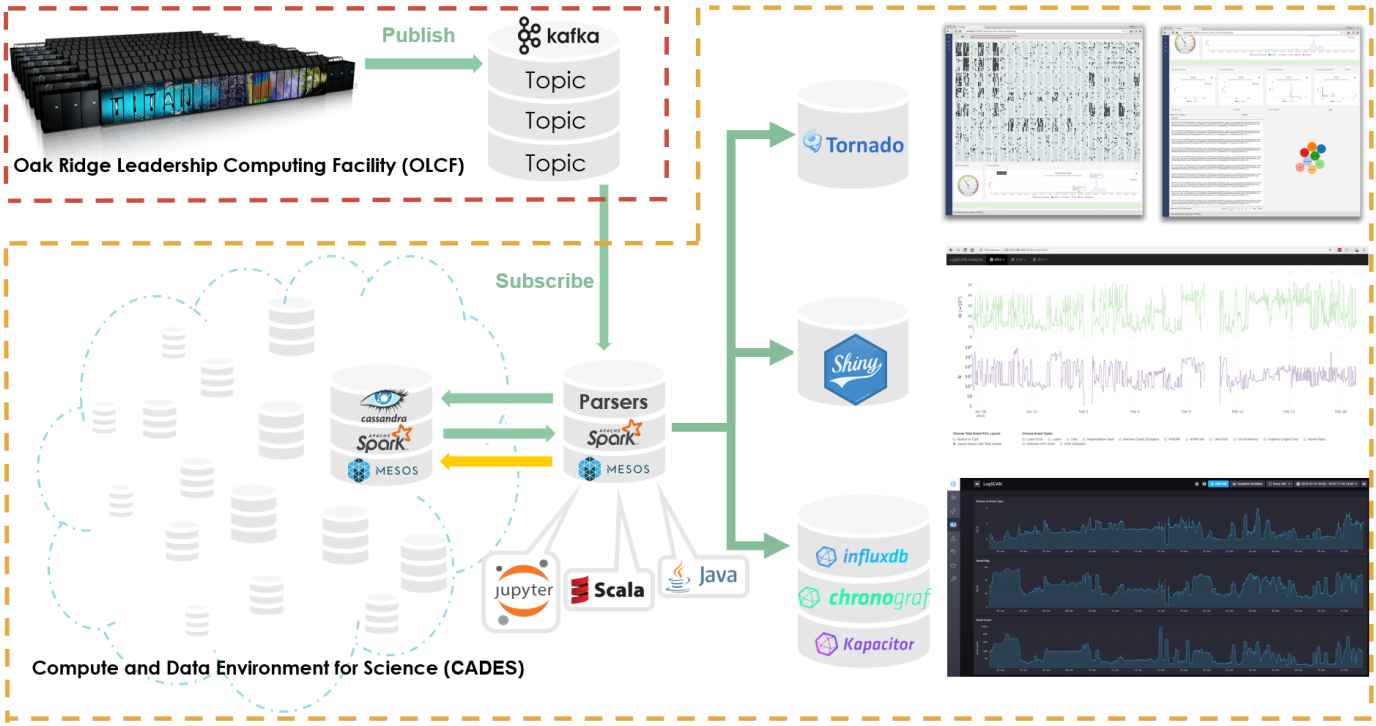

Fig. 2: A sketch of the architecture of LogSCAN. Two user facilities at ORNL, the Oak Ridge Leadership Computing Facility (OLCF) and the Computer and Data Environment for Science (CADES) provide hardware and software support for our work. Except for the Kafka servers which are hosted at OLCF, all the data portal, analysis and visualization servers are hosted on CADES through deployments of multiple OpenStack VMs.

of all reliability, availability and serviceability (RAS) events occurring on all the nodes and in various system components by leveraging powers of machine learning techniques and information theory; 2) *Application System Impact (ASI)* metric, which captures the state of all applications running on the system.

The SIE metric is used to make a sensitive diagnostic regarding whether different features, which are driving the evolution of the system status, act independently or are highly correlated. Given a general data layout of RECORD vs. FEATURE in a 2D table at each time instant, we proceed by computing the relative variances among all principal components [5] of the data table. The resulting variance probability vector could be further compressed with a Shannon entropy [6] to obtain the SIE at the given time instant.

For consolidating the impact of user applications on system state [7], ASI is used to identify instants of time, where the applications (or users) may be causing RAS events as compared to instants where issues in the system may be impacting execution of applications. ASI is based on the formation of a 1D vector which represents the sum of all events seen across all different applications executing in a given time window. The sparsity of this vector is quantified and is used to distinguish between cases where only few applications are generating bulk of the events from cases when most applications are impacted by various events.

III. Architecture of the Real-Time Monitoring Framework

In this work, we extend our log analysis framework as an end-to-end streaming process pipeline. Depicted in Figure 2, the pipeline starts with data sources which are provided by a subscription service to multiple Kafka topics published by OLCF. The raw log entries retrieved from these topics are parsed at the LogSCAN portal server and subsequently the emerged events are either stored in a Cassandra database for offline analytics or fed directly into various Spark Streaming Processors which simultaneously compute, according to the ultimate requirements from end users, the SIE or ASI metrics. The application interfaces are established through multiple APIs such as long-term Scala/Java apps or interactive Jupyter notebooks. And all analysis job submissions are orchestrated by a Mesos cluster management system. Within the analysis pipeline, we connect the output from the Spark Streaming Processors with various hosts of data storage/visualization. Especially convenient for online analytics and visualization of time series, we create a service stack around an InfluxDB database and choose Chronograf as the visualization terminal.

The InfluxDB data model is designed with such a principle that, as many as possible, categorical meta-data should be designated in a tag-set instead of a field-set for any ”measured” point. For example, in the case of SIE, meta-data which differentiate computations with various Data_Layout,

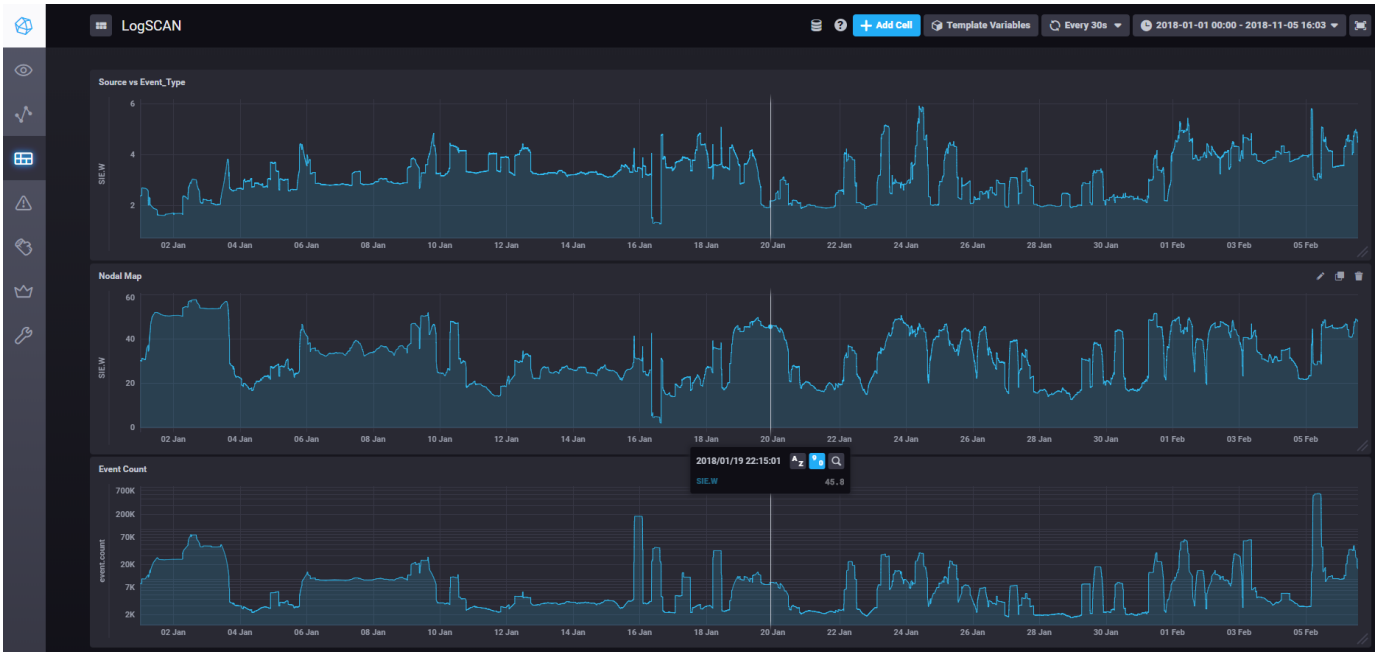

Fig. 3: An illustration of the real-time monitoring dash board for SIEs (in two data layouts) and total event counts. For a simple illustrative purpose, we only show the historical log data from Titan from January 1^{st} to February 6^{th}, 2018.

Time_Window, Time_Resolution, and Event_Type are stored as four tags for measurements which contains only two fields, the Shannon entropy and SIE itself. The advantage of our choice for the database schema is that all meta-data selected in the SIE will be implicitly indexed by InfluxDB and will dramatically accelerate general database queries.

Since all time series is stored in the InfluxDB-compliant format, we leverage Chronograf (one of the authentic components in the TICK Stack) as the visualization backend (shown with an example in Figure 3) due to its simple integration with InfluxDB and powerful visualization capability. Serving as an auxiliary component of the monitoring system, we are also testing an alarming framework built using the streaming engine Kapacitor out of the TICK Stack.

IV. Conclusions and Future Work

The metrics and their evaluation in real-time provides the opportunity to utilize pattern recognition, causality analysis in conjunction with other independent system metrics, and similarity analysis as a dynamic system. We believe that the application of this framework will create much needed insight into the evolution of HPC system status over time and its use can be easily extended to other setups which generate time series data.

Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Resilience for Extreme Scale Supercomputing Systems Program, with program manager Lucy Nowell, under contract number DE-AC05-00OR22725.

This work was also supported by the Compute and Data Environment for Science (CADES) facility and the Oak Ridge Leadership Computing Facility (OLCF) at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC for the U.S. Department of Energy (under the contract No. DE-AC05-00OR22725).

References

[1] B. H. Park, Y. Hui, S. Boehm, R. A. Ashraf, C. Engelmann, and C. Layton, "A big data analytics framework for HPC log data: Three case studies using the titan supercomputer log," in *Proceedings of the* 19^{th} *IEEE International Conference on Cluster Computing (CLUSTER) 2018*, Belfast, UK, Sept. 2018.

[2] Y. Hui, B. H. Park, and C. Engelmann, "A comprehensive informative metric for analyzing HPC system status using the LogSCAN platform," in *Proceedings of the* 31^{st} *International Conference on High Performance Computing, Networking, Storage and Analysis (SC) Workshops 2018*, Dallas, TX, USA, November 2018.

[3] J. Brandt, K. Devine, A. Gentile, and K. Pedretti, "Demonstrating improved application performance using dynamic monitoring and task mapping," in *2014 IEEE International Conference on Cluster Computing (CLUSTER)*, Sept 2014, pp. 408–415.

[4] O. Tuncer, E. Ates, Y. Zhang, A. Turk, J. Brandt, V. J. Leung, M. Egele, and A. K. Coskun, "Diagnosing performance variations in hpc applications using machine learning," in *High Performance Computing*, 2017, pp. 355–373.

[5] K. Pearson F.R.S, "LIII. On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[6] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[7] R. A. Ashraf and C. Engelmann, "Analyzing the impact of system reliability events on applications in the Titan supercomputer," in *Proceedings of the* 31^{st} *International Conference on High Performance Computing, Networking, Storage and Analysis (SC) Workshops 2018*, Dallas, TX, USA, November 2018.