LogDiver+: Advanced Analytics for HPC Errors and Failures

Saurabh Jha
Department of Computer Science
University of Illinois at Urbana - Champaign
Illinois, USA
sjha8@illinois.edu
http://web.engr.illinois.edu/~sjha8/

Keywords:
Resiliency
Fault Tolerance
Log Analysis

Background

Many important scientific studies such as weather forecasting, drug discovery etc. can be carried out only on extreme scale High Performance Computing (HPC) systems. There is a need to build exascale computers to further progress scientific studies and meet the ever growing demands of computing power. One of the critical problems – if not the most critical problem – in reaching exascale computing goal by the end of the decade is "designing fault tolerant applications and systems that can reach sustained petaflops to exaflops of performance" [7]. Given the complexity and scale of these systems, frequency of errors and failures have become an important issue to handle. In past few decades, "peak performance is growing faster than resilience". Although, today's supercomputer such as Tianhe -2 can attain a peak performance of 33.86 petaflops, presently there are no application that can sustain even 10% of this peak performance for over an hour. It is because such systems require coordination among millions of devices and in some way all these devices are competing to fail. This is only going to be more acute in exascale systems. Hence, there is a need for building systems and application at scale that can compute through errors and failures. One way to achieve this goal is to study historical extreme scale systems logs collected from various supercomputing sites, which are generally in order of petabytes, to build models to understand the reasons for failure, detect failures in advance and take proactive measures to prevent application failures. These logs have all the four characteristics of "big data" – velocity, veracity, volume and variety [5]. The volume and variety of data will ensure the correctness of the built models whereas veracity and velocity present opportunities for cross validating and updating models on the fly.

Problem Statement

Historically, computing systems fault tolerance is realized through circuit hardening, error detection and correction, error detection and retry, or auto-failover designs with N+1 or N+2 redundant circuits (N being a positive integer, 1, 2, … with +1 meaning 1 unit of extra circuit). If errors cannot be recovered in that manner, then the application fails, and after machine repair or re-configuration (or, in the case of a soft/transient error, the machine may remain unchanged), the application may be restarted from the beginning, or from a previously recorded checkpoint. Improved fault tolerance comes from detecting and auto-correcting a large percentage of errors.

In some cases, observing the rate of corrected errors through offline analysis can give system administrators warning of an impending failure. Such a warning may be followed by preventative actions, such as check pointing, replacement of failing components during planned down times, job-scheduling around ailing nodes, or proactive job migration [1, 2, 3, and 4].

Prior research have shown the importance of study of error logs [5, 6, 8] to find system and application failure patterns in order to take proactive actions to contain error propagation. However, all these studies were done for systems that were 10x – 100x times smaller than current systems. Further, these studies are scattered and does not provide means to assist different techniques to strengthen and advance the study of resiliency in general.

Error logs are gold mine of information that can assist in building more resilient systems but there must be a unified way to represent the data so as to enable comprehensive resiliency study and validation of resiliency mechanisms on a live system. Modern supercomputer systems generate a variety of information such as environment sensors, performance logs, and error logs that could be used for understanding the behavior of systems and applications. An in-depth analysis of these data allows uncovering fingerprints (e.g., in terms of patterns of system calls or messages exchanged between software components) corresponding to a correct and/or anomalous behavior application

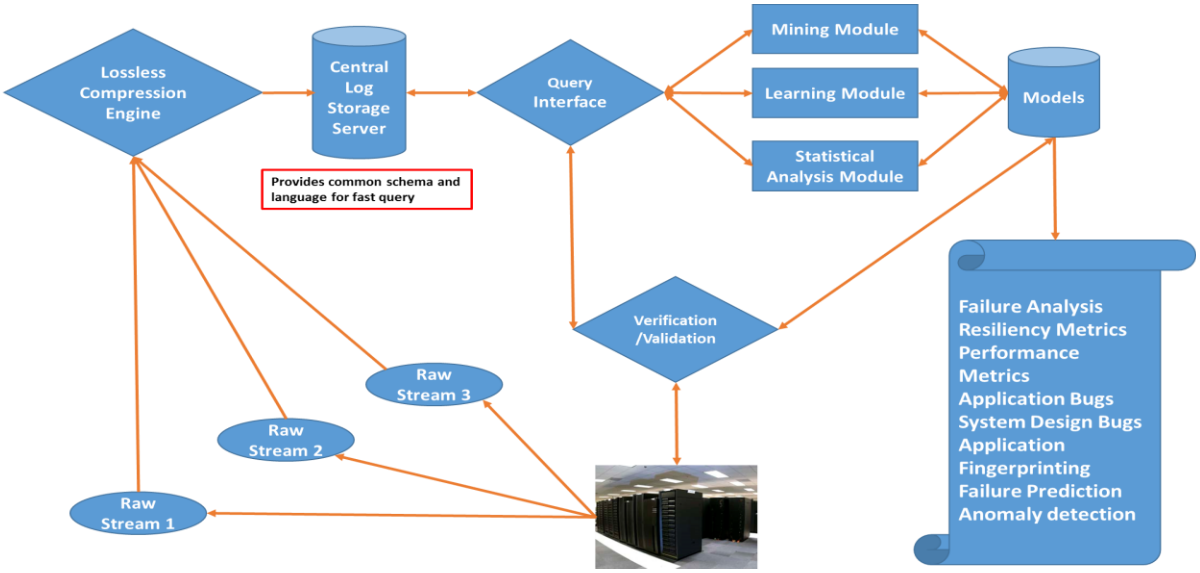

Figure 1 LogDiver+

Given the size of error logs, in order of petabytes for typical extreme scale systems, storage and processing of these logs have become a challenging problem. In order to address this challenge, there is a need for building unified advanced log analytics and storage engine, **LogDiver+** so as to facilitate resiliency studies and improve the resiliency of overall system and programs. Such a tool (see Figure 1) should be capable of:

1. Efficiently compress (lossless compression), store, process, query and mine petabytes of errors logs. This will require building common schema, language and parallel storage engine from scratch, specialized for resiliency study.

2. Provide flexible data structure and formats to support – quantification of resiliency and performance of systems and applications, finding system design problems / application bugs, modelling application failure path (fingerprinting) and building prediction models.
3. Provide mechanism to cross validate models/results through statistical approaches as well as domain knowledge (such as through fault injection).

Broader Impacts

Above mentioned approach encompasses techniques from big data systems, parallel programming, statistics and machine learning to assist in resiliency studies to support building exascale systems and beyond. It will help discover failure patterns, error propagation paths, error containment strategies etc. It will also help discover design and application bugs and bottlenecks which are hard to spot by a human expert. Further, such an understanding will help advances in the process of data collection, storage and analysis of logs in future supercomputers (and large scale clouds) for continuous resiliency monitoring.

References

[1] F. Capello, A. Geist, W. Gropp, S. Kale, B. Kramer, M. Snir, Toward Exascale Resilience: A 2014 Update. http://www.mcs.anl.gov/papers/P5147-0614.pdf

[2] C. Engelmann, G. R. Vallee, T. Naughton, S. L. Scott, Proactive fault tolerance using preemptive migration. In Proceedings of the 17th Euromicro International Conference on Parallel, Distributed, and network-based Processing (PDP) 2009.

[3] Litvinova, C. Engelmann, S. L. Scott, A proactive fault tolerance framework for high-performance computing. In Proceedings of the 9th IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN) 2010.

[4] http://www.ibmbigdatahub.com/infographic/four-vs-big-data

[5] Antonio Pecchia, Domenico Cotroneo, Zbigniew Kalbarczyk, Ravishankar K. Iyer: Improving Log-based Field Failure Data Analysis of multi-node Computing Systems. DSN 2011: 97-108

[6] Ravishankar K. Iyer, Luke T. Young, P. V. Krishna Iyer, “Automatic Recognition of Intermittent Failures: An Experimental Study of Field Data,” IEEE Trans. Computers 39(4): pp. 525-537, 1990.

[7] Shalf, John, Sudip Dosanjh, and John Morrison. "Exascale Computing Technology Challenges." High Performance Computing for Computational Science–VECPAR 2010. Springer Berlin Heidelberg, 2011. 1-25.

[8] Martino, C. D., Jha, S., Kramer, W., Kalbarczyk, Z., & Iyer, R. K., "LogDiver: A Tool for Measuring Resilience of Extreme-Scale Systems and Applications." Proceedings of the 5th Workshop on Fault Tolerance for HPC at eXtreme Scale. ACM, 2015.