On the Path to the Holy Grail:

Predicting onset of system failure with log files

R. Settlage, M. Marshall, K. Senthilvel, V. Agarwala, J. Akers
Advanced Research Computing
Virginia Polytechnic Institute and State University
Blacksburg, VA 24060, U.S.A
rsettlag@vt.edu
mimarsh2@vt.edu
skarthik@vt.edu
vijaykag@vt.edu
akers@vt.edu

R. Bendale, K. Robertson
Engility Corporation
Chantilly, VA 20151, U.S.A
rajiv.bendale@engilitycorp.com
Kimberly.Robertson@engilitycorp.com

Keywords—failure analysis,High Performance Computing

I. Extended Abstract

Scientific problems only get bigger. To supply the required compute capacity necessary to answer the cutting edge questions in science, engineering, analytics etc., HPC systems have continued to grow in size and complexity [1]. With the plateauing of CPU clock speeds, new architectures have emerged to create heterogeneous machines increasing overall system complexity. Individual component reliability remains high, but with thousands of nodes the mean time between failure of any node can be extremely short [2]. Job checkpointing at the frequency of single node failures is not feasible.

University HPC clusters have hundreds of nodes versus thousands of nodes and typically buy commodity components from large OEMs as opposed to experimental silicon e.g. BlueGene [3] systems. Fewer, more-commodity nodes leads to component failures being less of an issue when deploying new systems. Additionally, university HPC centers are typically tying together multiple legacy systems to shared networks and storage systems, which can cause unanticipated bottlenecks and show component failures. Previous work [4] [5][6] has focused on large systems at national labs. This work focuses on anomaly detection as a prelude to failure prediction at university-scale HPC centers. Further, we demonstrate the utility of anomaly detection as a diagnostic tool for identifying misconfigurations or performance issues using stale file handles as our target message.

We show system logs in aggregate can be analyzed to predict system states corresponding to normal, misconfiguration and hardware errors. Log messages, even informational ones, form a "heartbeat" for the system which can be used to classify the overall health of the system. Modeling approaches including machine learning, random forest trees, and hierarchical clustering are used to determine a "normal" state of ARC systems and detect anomalous "abnormal" states possibly leading to "error" states.

This work analyzes 4 months of system logs from ARC's latest HPC cluster, NewRiver [7], a 126 node, Intel Haswell based system interconnected with Mellanox EDR Infiniband released to Virginia Tech research community in August 2015. We also include logs from shared infrastructure systems and legacy clusters. We developed a scalable process to parse log files and calculate likelihood of the system slipping into an error state. The process has 3 general steps:

1. Generate a message dictionary from log data
2. Use unsupervised classification techniques to determine if a given time interval is "abnormal"
3. Identify the abnormal states as known error conditions using trained neural networks

To transform the textual log messages into a format suitable for processing, we create message dictionaries in which each message is classified as a given message type. Prior work [8] has started by manually categorizing messages and events using prior knowledge. We create a heuristic-based program for automatically building the message dictionary. This removes the time-consuming work of hand creating message types and allows for a system that can quickly adapt as message types change.

Given the message distribution for a given time sample e.g. 5 minutes, we determine the state of the system during that time interval. We use unsupervised classification techniques to group time samples into either "normal" or "abnormal/outlier" states for a given time period. For example, over a 3 week time period, we group every 5 minute time sample into either "normal" or "abnormal". By using an unsupervised learning method, we let the data group itself rather than using prior knowledge to pick specific data we know leads to an error state. This allows us to gain insight into log messages that we may have thought irrelevant and also detect abnormal states beyond the storage errors.

Once detected, supervised learning methods, neural networks, are used to identify abnormal states as specific known “error” states or simply “unknown” if no matches are found. The parameters used to train the neural networks are picked based on the best results of the unsupervised learning. For example, the time interval, which created the best correlations in the unsupervised step for our data, is used as the length of the training set. Again, we let the data dictate the model used to gain insights beyond our prior knowledge.

Given 4 months of system logs, we were challenged with developing a method for predicting system failures which impact user jobs. This poster presents an algorithm developed for classifying system state as normal or abnormal from log messages generated within a given time interval. When an abnormality is detected, we further calculate the likelihood that an error state will soon occur.

References

[1] R. Gioiosa, "Towards sustainable exascale computing," in VLSI System on Chip Conference (VLSI-SoC), 2010 18th IEEE/IFIP, sept. 2010.

[2] F. Cappello, A. Geist, and W. Gropp. Toward Exascale Resilience: 2014 update. Supercomputing Frontiers and Innovations, 1(1):5–28, 2014.

[3] A. Gara et. al, “Overview of the Blue Gene/L system architecture,” IBM Journal of Research and Development, vol 49, no. 2/3, pp. 195 – 212, March/May 2009.

[4] A. Gainaru, F. Cappello, M. Snir, and W. Kramer. Failure prediction for HPC systems and applications: Current situation and open issues. International Journal of High Performance Computing Applications, 27(3):273–282, July 2013.

[5] Y. Liang, Y. Zhang, H. Xiong and R. Sahoo, "An adaptive semantic filter for blue gene/L failure log analysis[C]", Proceedings of the Third International Workshop on System Management Techniques, Processes, and Services (SMTPS), March 2007.

[6] Y. Liang, Y. Zhang, A. Sivasubramaniam, M. Jette and R. Sahoo, "Blue Gene/L Failure Analysis and Models," Proc. Int',l Conf. Dependable Systems and Networks (DSN), 2006.

[7] http://www.arc.vt.edu/computing/newriver

[8] Z. Zheng, Z. Lan, B. Park, and A. Geist, "System Log Pre-processing to Improve Failure Prediction". Proc. Int',l Conf. Dependable Systems and Networks (DSN), 2009.