Pin-pointing Node Failures in HPC Systems

[Extended Abstract]

Anwesha Das
North Carolina State University
adas4@ncsu.edu

Frank Mueller
North Carolina State University
fmuelle@ncsu.edu

Paul Hargrove
Lawrence Berkeley National Laboratory
phhargrove@lbl.gov

Eric Roman
Lawrence Berkeley National Laboratory
ERoman@lbl.gov

ABSTRACT

Automated fault prediction and diagnosis in HPC systems needs to be efficient for better system resilience. With increasing scalability required for exascale, accurate fault prediction aiding in quick *remedy* is hard. With changing supercomputer architectures, distilling fault data from the noisy raw logs requires substantial efforts. Predicting node failures in such voluminous system logs is challenging.

To this end, we investigate an interesting way to pinpoint node failures in such supercomputing systems. Our study on *Cray* system data with automated machine learning tools suggests that specific patterns of event messages on node *unavailability* can be indicator to node failures. This data extraction coupled with system and job data correlation helps in devising a methodology to predict node failures and their location over a specific time frame. This work aims to enable broader applicability for a generic fault prediction framework.

Keywords

HPC Logs; Cray; Failure Prediction; Node Failures

1. INTRODUCTION

HPC resilience has been researched extensively and prior work has studied system logs in the context of fault detection and prediction. While most works have focused on *BlueGene* system [6, 11, 3, 8], the most popular HPC systems these days are *Cray* machines amongst the top supercomputers [2].

From log data analysis to root cause diagnosis across various levels (hardware, system, application) researchers have studied failure manifestations in HPC systems and devised ways to improve precision and recall rates [7]. In-spite of this body of work on resilience we feel further investigation is required for the following reasons:

DOI: 10.1145/1235

- Existing work performs prediction and diagnosis without emphasizing real-time mitigations. *Pin-pointing nodes* which will fail in the future well ahead in time to pro-actively recover from performance disruptions still remains a challenge. Choosing an optimal learning window interval and *lead time* are important considerations for successful prediction of *node* failures.

- Most prior works [11, 12, 5] use the same training data for future predictions over a long time frame. As hinted in Gainaru et al. [7], correlations found off-line are not changed, which limit the approach when using a short training set for long future time window. This limitation makes prediction unrealistic when used on production systems. We should investigate further *dynamic learning techniques* and *online prediction* to improve prediction accuracy like Gu et al. [8].

- There exist unpredictable failures [7]. Understanding Cray systems to see existence of such cases where *correlation extraction* is hard from system logs needs to be studied.

- Most resilience work [6, 12, 3, 5] features case studies of systems, which are decommissioned and not in use like the BlueGene system Mira/Ranger etc. Currently used systems require further studies to understand requirements of resilience.

- Validation of predicted faults is done through comparison with event logs or by discussing with the system administrators. Relying on manual human expertise or system administrator's knowledge is difficult at times. Is there any better validation scheme for good prediction accuracy?

Recent work [10, 9, 4] gives helpful insights on Cray data studying Titan [1]. This work shows interesting ways to pin-point failures by phrase extraction and time-based event correlation.

2. SOLUTION APPROACH

Data preprocessing considering important features across various directories and files in Cray logs is challenging since identifying determining factors of failures is not straight forward. Additionally, there exists time slack between service

nodes, Job schedulers (like SLURM/TORQUE) and compute nodes making time-based correlation painful. In the light of these conditions, our work focuses on identifying node failures and figuring out machine learning tools that can automate failure detection from raw logs. Our work relies on data from the Cray platforms *Edison* and *Hopper* available at NERSC.

In our proof-of-concept design and implementation, we investigate three directions that can facilitate pin-pointing node (service/compute) failures: distinction between normal shutdown versus nodes abnormally going down, figuring out how to leverage time-series based error messages and phrase extraction methods to identify failures and, correlation of system data with job data to detect faulty nodes. While PCA (principal component analysis) is hard for the kind of text-data needed to be parsed, continuous time LDA (Latent Dirichlet Allocation) based probability analysis like unsupervised key-phrase extraction methods can help identify required text phrases. Mass service node shutdowns are common compared to a single compute node failure in a chassis. Identifying those rare compute node failures by relating job ids, which are *re-run* several times on those faulty nodes, with boot messages indicating a node's status can pin-point node failures. Our work derives such hints and insights from Cray data, which is vital in the context of fault prediction and subsequent proactive resilience actions.

3. CONCLUSION

Our findings reveal that service node reboots and maintenance related shutdowns are frequent, compute node failures because of faults are relatively rare. While boot messages, node health information and console messages can together extract node state and fault, additional job state information running on that node could facilitate identifying node failures. Since the focus is on *temporal* text mining rather than number based covariance analysis, leveraging unsupervised machine learning techniques like LDA, DTM (Dynamic Topic Model-ling) and similar phrase extraction methods could be a viable research direction.

4. REFERENCES

[1] https://www.olcf.ornl.gov/titan/.

[2] http://www.top500.org/featured/top-systems/.

[3] E. Berrocal, L. Yu, S. Wallace, M. E. Papka, and Z. Lan. Exploring void search for fault detection on extreme scale systems. In *Cluster Computing (CLUSTER), 2014 IEEE International Conference on*, pages 1–9. IEEE, 2014.

[4] J. Brandt, A. Gentile, C. Martin, J. Repik, and N. Taerat. New systems, new behaviors, new patterns: Monitoring insights from system standup. In *Cluster Computing (CLUSTER), 2015 IEEE International Conference on*, pages 658–665. IEEE, 2015.

[5] S. Fu and C.-Z. Xu. Exploring event correlation for failure prediction in coalitions of clusters. In *Supercomputing, 2007. SC'07. Proceedings of the 2007 ACM/IEEE Conference on*, pages 1–12. IEEE, 2007.

[6] X. Fu, R. Ren, S. A. McKee, J. Zhan, and N. Sun. Digging deeper into cluster system logs for failure prediction and root cause diagnosis. In *Cluster Computing (CLUSTER), 2014 IEEE International Conference on*, pages 103–112. IEEE, 2014.

[7] 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, 2013.

[8] J. Gu, Z. Zheng, Z. Lan, J. White, E. Hocks, and B.-H. Park. Dynamic meta-learning for failure prediction in large-scale systems: A case study. In *Parallel Processing, 2008. ICPP'08. 37th International Conference on*, pages 157–164. IEEE, 2008.

[9] S. Gupta, D. Tiwari, C. Jantzi, J. Rogers, and D. Maxwell. Understanding and exploiting spatial properties of system failures on extreme-scale hpc systems. In *Dependable Systems and Networks (DSN), 2015 45th Annual IEEE/IFIP International Conference on*, pages 37–44. IEEE, 2015.

[10] D. Tiwari, S. Gupta, and S. S. Vazhkudai. Lazy checkpointing: Exploiting temporal locality in failures to mitigate checkpointing overheads on extreme-scale systems. In *Dependable Systems and Networks (DSN), 2014 44th Annual IEEE/IFIP International Conference on*, pages 25–36. IEEE, 2014.

[11] L. Yu, Z. Zheng, Z. Lan, T. Jones, J. M. Brandt, and A. C. Gentile. Filtering log data: Finding the needles in the haystack. In *Dependable Systems and Networks (DSN), 2012 42nd Annual IEEE/IFIP International Conference on*, pages 1–12. IEEE, 2012.

[12] Z. Zheng, L. Yu, Z. Lan, and T. Jones. 3-dimensional root cause diagnosis via co-analysis. In *Proceedings of the 9th international conference on Autonomic computing*, pages 181–190. ACM, 2012.