Desh: Deep Learning for HPC System Health Resilience

Extended Abstract

Anwesha Das
North Carolina State University
adas4@ncsu.edu

Abhinav Vishnu
Pacific Northwest National Laboratory
abhinav.vishnu@pnnl.gov

Charles Siegel
Pacific Northwest National Laboratory
charles.siegel@pnnl.gov

Frank Mueller
North Carolina State University
fmuelle@ncsu.edu

ABSTRACT

HPC systems are well known to endure service downtime due to increasing failures. With enhancements in Supercomputing architectures and design, enabling resilience is extremely challenging due to component scaling and absence of well defined failure indicators. Supercomputing system logs are notorious to be complex and unstructured. Efficient fault prediction to enable proactive recovery mechanisms is the need of the hour to make such systems more robust and reliable. This work addresses such faults in computing systems using a recurrent neural network based technique called LSTM (long short-term memory).

We present our framework *Desh*[1]: *De*ep Learning for HPC *S*ystem *H*ealth, which entails a procedure to diagnose and predict failures with acceptable lead times. Desh indicates prospects of indicating failure indicators with enhanced training and classification for generic applicability to other systems. This deep learning based framework gives interesting insights for further work on HPC system reliability.

KEYWORDS

LSTM, Failure Prediction, Classification, HPC, Data Mining

1 INTRODUCTION

HPC systems suffer from various kinds of failures at the hardware, software and application levels. While some failures are critical and are obvious to detect such as kernel panics, most anomalies are not easy to track. What component will fail and how will it impact the system is not known ahead of time. Bautista-Gomez et al. [3] discusses the spatial and temporal analysis of DRAM memory errors in HPC systems. System data and job logs contain various events and messages related to the entire supercomputing system, extracting useful information pertaining to the impending failures for newer systems such as the Cray architectures needs further research. This work attempts to address failure prediction in HPC system by exploiting deep learning. Recently, several researchers have leveraged machine learning for the purpose of anomaly detection in large scale systems. Oliner et al. [13] propose Nodeinfo: an unsupervised alert detection system using ideas of information entropy and binary scoring. Xie et al. [17] assessed the Lustre file system of the Titan Supercomputer to propose a statistical regression approach, which predicts output performance in petascale file systems. Chilimbi et al. [4] propose Adam, a scalable deep learning training system. While deep learning has been investigated extensively in the areas of vision and speech recognition, its efficacy in the context of fault prediction and localization for HPC systems is unknown. This paper takes a step in that direction. The generic accuracy of prediction using deep learning over diverse HPC systems is yet to be investigated. Nevertheless, our analysis provides an interesting solution paradigm to handle text logs to predict anomalies. This can provide useful insights to the HPC community in general.

2 BACKGROUND

Recurrent neural networks (RNNs) have the power to predict future data based on sequences of past data. LSTMs (long short-term memory) have been particularly used for long term as well as short term data dependencies when chains of events pertaining to a domain have been provided as the input. Supercomputing logs have unstructured textual data with both short-term failures (e.g. kernel crash in 20 seconds) as well as long term failures (e.g. link control block failure in 5 minutes), irrespective of the root causes of such failures. Moreover, time-stamped logs have diverse events logged in the granularity of seconds, and patterns evolve over varying intervals of time. The logs contain phrases with anomalies interspersed with considerable amount of noise and benign events. The question is how do we analyse the data and efficiently leverage LSTM to predict future phrases? Can we have high prediction accuracy based on the expert labelled ground truth? Apart from trying to seek the above answers, our work differs from the prior state-of-the-art in the following ways:

- Prior log analysis techniques have studied various event correlation methods [18], time coalition techniques [7] and log parsing method evaluation [10]. Our work uses the phrases specific to diverse components of the logs to handle failure prediction. Our work emphasizes *semantics of phrases* and their appearances in the chain of events over time.
- Recent failure prediction approaches such as Hora [16] and Nodeinfo [13] either do not stress on lead time or use fault injection and synthetic data for evaluation or do not consider the semantic information in the log entries. Our work intends to highlight prediction accuracy with acceptable lead times when deep learning techniques are applied on the real textual logs of a contemporary Cray system.

Past solutions based on PCA/ICA (principal/independent component analysis) [12], probabilistic model, markov chain and decision tree, worked for systems with comparatively more structured logs, aiding in feature extraction and offline anomaly detection. They

[1] *Desh* means *Native Land* in the Indian Language

are inefficient when it comes to unstructured text data mining with time constraints. Prevalent approaches such as Support Vector Machines (SVMs) [9] and sequence mining [8] either require complex feature extraction or are unable to capture long term dependencies, making systems intractable with scale. Very recently Coates et al. [5] demonstrated that large scale training can be done through deep learning on HPC infrastructures with acceptable classification performance and scalable efficiency. LSTM works well for time sensitive data. It can unlearn and relearn over time, making it a preferable choice over other RNNs such as logistic regression and multilayer perceptron (MLP).

3 SOLUTION PARADIGM

We present a framework called Desh: *De*ep Learning for HPC *S*ystem *H*ealth, to predict failures. Our idea is to analyse system logs efficiently using LSTM for quality failure prediction. Since we deal with semantic information buried in textual phrases, our work does not group log entries based on temporal locality alone. Pecchia et al. [14] discusses that grouping events based on predetermined time threshold performs badly. Additional consideration of likelihood of entries improves field data analysis. Di Martino [6] uses MTW (multiple time windows) heuristic to group supercomputing error logs. Desh groups phrases from the logs into three categories: safe, error and unknown. *Safe* represents the benign phrases, which are definitely not related to any anomaly. *Error* refers to the fatal/critical phrases, which are definitely indicative of some anomaly. The *Unknown* tag is given to those phrases, which may or may not be indicative of any anomaly or are simply information not indicating any specific event. This phrase grouping is based on expert guidance and filtered labelling. The phrases from every text file are then scrubbed into static and dynamic contents to identify the constant message type, separating it from the variable component. Once the constant messages are extracted they are encoded to a uniquely identifiable number. The encoded data is further processed to form a time-stamp augmented sequence vector of messages. Desh feeds this vector to the LSTM, and outputs phrases that are likely to occur in the future for certain specified time bins. These predicted phrases are then classified into one of the three categories mentioned earlier (safe, error, unknown). In this way, Desh helps to induce how far ahead in time are erroneous phrases likely to occur. The idea of event block detection described by Baseman et al. [2] is done in similar ways by several researchers who are focusing on tokenizing of unstructured text. Desh aims to scrub the phrase containing multiple entries with static and dynamic contents into a single uniquely identifiable phrase type pertaining to an event or message. Desh's failure prediction is similar to Hora [15, 16], however the latter uses the Weka package for all the filters of classification and prediction on BlueGene/L logs. Desh uses ideas of temporal phrase mining along with deep learning over Cray XC logs to predict failures.

4 CONCLUSIONS

LSTM has been used in the context of engine prediction condition (Aydin et al. [1]) and text embeddings for Natural Language Processing (NLP) (Johnson et al. [11]) apart from its extensive usage in vision and speech. Our prototype, Desh, provides a useful technique to process Cray logs using LSTM for efficient failure prediction. Desh successfully predicts impending phrases which can facilitate correct failure indication. Our future plan is to enable sufficient lead time for higher prediction accuracy and investigate opportunities for performance optimization. This work may inspire interesting insights for characterising deep learning technique when applied to the unstructured HPC system logs for enhanced reliability.

REFERENCES

[1] Olgun Aydin and Seren Guldamlasioglu. 2017. Using LSTM networks to predict engine condition on large scale data processing framework. In *Electrical and Electronic Engineering (ICEEE), 2017 4th International Conference on*. IEEE, 281–285.

[2] Elisabeth Baseman, Sean Blanchard, Zongze Li, and Song Fu. 2016. Relational Synthesis of Text and Numeric Data for Anomaly Detection on Computing System Logs. In *Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on*. IEEE, 882–885.

[3] Leonardo Bautista-Gomez, Ferad Zyulkyarov, Osman Unsal, and Simon McIntosh-Smith. 2016. Unprotected computing: a large-scale study of DRAM raw error rate on a supercomputer. In *High Performance Computing, Networking, Storage and Analysis, SC16: International Conference for*. IEEE, 645–655.

[4] Trishul M. Chilimbi, Yutaka Suzue, Johnson Apacible, and Karthik Kalyanaraman. 2014. Project Adam: Building an Efficient and Scalable Deep Learning Training System. In *11th USENIX Symposium on Operating Systems Design and Implementation, OSDI '14, Broomfield, CO, USA, October 6-8, 2014.* 571–582.

[5] Adam Coates, Brody Huval, Tao Wang, David Wu, Bryan Catanzaro, and Ng Andrew. 2013. Deep learning with COTS HPC systems. In *International Conference on Machine Learning*. 1337–1345.

[6] Catello Di Martino. 2013. One size does not fit all: Clustering supercomputer failures using a multiple time window approach. In *International Supercomputing Conference*. Springer, 302–316.

[7] Catello Di Martino, Marcello Cinque, and Domenico Cotroneo. 2012. Assessing time coalescence techniques for the analysis of supercomputer logs. In *Dependable Systems and Networks (DSN), 2012 42nd Annual IEEE/IFIP International Conference on*. IEEE, 1–12.

[8] Xiaoyu Fu, Rui Ren, Sally A McKee, Jianfeng Zhan, and Ninghui Sun. 2014. Digging deeper into cluster system logs for failure prediction and root cause diagnosis. In *Cluster Computing (CLUSTER), 2014 IEEE International Conference on*. IEEE, 103–112.

[9] Errin W Fulp, Glenn A Fink, and Jereme N Haack. 2008. Predicting Computer System Failures Using Support Vector Machines. *WASL* 8 (2008), 5–5.

[10] Pinjia He, Jieming Zhu, Shilin He, Jian Li, and Michael R Lyu. 2016. An evaluation study on log parsing and its use in log mining. In *Dependable Systems and Networks (DSN), 2016 46th Annual IEEE/IFIP International Conference on*. IEEE, 654–661.

[11] Rie Johnson and Tong Zhang. 2016. Supervised and semi-supervised text categorization using LSTM for region embeddings. In *International Conference on Machine Learning*. 526–534.

[12] Zhiling Lan, Ziming Zheng, and Yawei Li. 2010. Toward automated anomaly identification in large-scale systems. *IEEE Transactions on Parallel and Distributed Systems* 21, 2 (2010), 174–187.

[13] Adam J Oliner, Alex Aiken, and Jon Stearley. 2008. Alert detection in system logs. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, 959–964.

[14] Antonio Pecchia, Domenico Cotroneo, Zbigniew Kalbarczyk, and Ravishankar K Iyer. 2011. Improving log-based field failure data analysis of multi-node computing systems. In *Dependable Systems & Networks (DSN), 2011 IEEE/IFIP 41st International Conference on*. IEEE, 97–108.

[15] Teerat Pitakrat, Jonas Grunert, Oliver Kabierschke, Fabian Keller, and André Van Hoorn. 2014. A framework for system event classification and prediction by means of machine learning. In *Proceedings of the 8th International Conference on Performance Evaluation Methodologies and Tools*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 173–180.

[16] Teerat Pitakrat, Dušan Okanović, André van Hoorn, and Lars Grunske. 2017. Hora: Architecture-aware online failure prediction. *Journal of Systems and Software* (2017).

[17] Bing Xie, Yezhou Huang, Jefrey S Chase, Jong Youl Choi, Scott Klasky, Jay Lofstead, and Sarp Oral. 2017. Predicting Output Performance of a Petascale Supercomputer. In *Proceedings of the 26th International Symposium on High-Performance Parallel and Distributed Computing*. ACM, 181–192.

[18] Chen Zhuge and Risto Vaarandi. 2017. Efficient Event Log Mining with LogClusterC. In *2017 IEEE 3rd International Conference on Big Data Security on Cloud (BigDataSecurity)*. 261–266.