Aarohi: Automaton-based Low-cost Online Failure Prediction

Extended Abstract

Anwesha Das, Frank Mueller
North Carolina State University
{adas4,fmuelle}@ncsu.edu

ABSTRACT

Large-scale production systems suffer from failures, wasting compute capacity and power. Both cloud computing and supercomputing system logs have been investigated in the context of anomaly detection. The existing data mining solutions are not sufficiently fast for real-time anomaly detection. Machine learning (ML)-based training can produce high accuracy but the detection scheme needs to be enhanced with rapid checkers to infer anomalies quickly. This work addresses online anomaly detection in computing systems, using automaton-based timely event transitions.

We present our framework *Aarohi*[1], which describes an efficient methodology for fast failure detection. Aarohi is generic, scalable, making it suitable for a real-time detector.

1 INTRODUCTION

In the recent past, researchers have solved pertinent problems related to anomaly detection for system reliability. Unstructured log mining-based failure characterization [2] for contemporary supercomputing systems is undergoing extensive research. With the upcoming exascale era, scaling the proposed solutions to work efficiently in real-time is critical. The efforts invested in unveiling the wealth of information from ML- and deep learning (DL)-based studies [1, 3] will truly pay off, when we take steps to build realistic frameworks improving system availability. ML-based solutions have been found effective for offline training, and detection. Performance optimized training has produced high recall rates and accuracy [1]. However, from the trained failure indicators, inferring impending failures from the new test data, may not be fast enough to aid real-time detection. DeepLog and Cloudseer [1, 4] incurs 3.48 and 1.81 millisecs respectively, to check a single log entry for online detection using techniques of Long Short-Term Memory (a recurrent neural network model) and task automatons. Log messages are generated at the granularity of μsecs. Can we detect anomalies any faster? Will that reduce the accuracy? To what extent? This necessitates research efforts to build online detectors that are adaptive, generic and fast. We propose to build a tool that automates the transition from learned event failure chains to a fast real-time parser of log event streams to predict failures.

2 MOTIVATION

While researchers have proposed failure prediction solutions, most of them cannot be used online [3] to take proactive recovery actions (e.g., job migration, process cloning). Some of the hurdles for real-time failure prediction in large-scale systems are:

1. The pace of analyzing event logs by the detector should be abreast with the pace at which the system generates it.
2. The existing ML-based schemes are effective offline trainers, but their detection rate is unsuitable for real-time processing speed.
3. An online detector should be reusable with the evolving event patterns, accommodating software/logging paradigm upgrades with minimum overhead, not receding its efficacy over time.

Aarohi intends to contribute a fully automated unsupervised parser from a DL-based solution. It aims to perform significantly better failure prediction via novel event stream parsing of failure chains.

3 SOLUTION PARADIGM

The main idea of our proposed Aarohi framework is to use deterministic finite automata (DFA) for generating rules with the failures from the DL-based training. Aarohi tokenizes the test data into timestamps & event messages, and checks them against the predefined rules to evaluate whether the sequence of events indicates a failure. The prototype is built using flex/bison parsers. The flex scanner tokenizes the input message using regular expressions and sends to the bison parser. The bison parser (aka failure checker) entails the grammar rules formulated from the failure chains. The rules are formulated based on the message sequence and the time difference between the two events. The latter captures the contextual relevance of events indicating failures. The parser compares the tokens with the rules to detect failures. In-face of software or logging paradigm upgrades, updating the rules with new event ids/operators suffices, without changing the entire parser workflow. The rules can be recursively applied for similar event sequences with variable prefixes. Aarohi efficiently detects failure chains with diverse event change points in ≈1 millisecs and an event phrase in ≈70 μsecs. Aarohi extends to multi-instance DFAs to check multiple rules simultaneously, with the ability to start/stop different failure chains based on what event phrase is observed next.

4 CONCLUSIONS

DFAs and finite-state machines have been leveraged for cognition and automated semantic recognition by computational linguists. Static rule-based bug checkers used in distributed systems/software engineering are not time-sensitive failure predictors. Aarohi can be used for both HPC and cloud systems to predict failures online, with acceptable lead times.

REFERENCES

[1] M. Du, F. Li, G. Zheng, and V. Srikumar. 2017. DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning. In *CCS*.
[2] S. Gupta, T. Patel, C. Engelmann, and D. Tiwari. 2017. Failures in Large Scale Systems: Long-term Measurement, Analysis, and Implications. In *SC*.
[3] Z. Lan, Z. Zheng, and Y. Li. 2010. Toward automated anomaly identification in large-scale systems. *IEEE TPDS*.
[4] X. Yu, P. Joshi, J. Xu, G. Jin, H. Zhang, and G. Jiang. 2016. Cloudseer: Workflow monitoring of cloud infrastructures via interleaved logs. *SIGOPS OS Review*.

[1] *Aarohi* means *ascending* in *Sanskrit* Language. It personifies the gradual event wise progression towards successful failure prediction.