A LIGHTWEIGHT DYNAMICALLY EVOLVING ONLINE FAILURE PREDICTION APPROACH

Mohammed Ibrahim.M
Software Engineering
Sona College of technology
Salem, India
mniibrahim1@outlook.com

Reena Devi.R
Computer Science &Engineering
Sona College of technology,
Salem, India
reenadevi@sonatech.ac.in

ABSTRACT

Online failure prediction approaches aim to predict the manifestation of failures at runtime before the failures actually occur. Existing approaches generally refrain themselves from collecting internal execution data, which can further improve the prediction quality. One reason behind this general trend is the runtime overhead incurred by the measurement instruments that collect the data. Since these approaches are targeted at deployed software systems, excessive runtime overhead is generally undesirable. In this work we conjecture that large cost reductions in collecting internal execution data for online failure prediction may derive from pushing the substantial parts of the data collection work onto the hardware. To test this hypothesis, we present a lightweight online failure prediction approach, called Seer, in which most of the data collection work is performed by fast hardware performance counters. The hardware-collected data is augmented with further data collected by a minimal amount of software instrumentation that is added to the systems software. In our empirical evaluations conducted on three open source projects, Seer performed significantly better than other related approaches in predicting the manifestation of failures.

KEYWORDS: Seer, Fault Prediction, System Spectra, Hybrid Spectra.

1. INTRODUCTION

Software systems do fail in the field by following this pragmatic line of thought, many online failure prediction approaches have been developed to predict the manifestation of failures at runtime, i.e., while the system is running and before the failures occur, so that preventive measures, such as system reboots, or protective measures, such as check pointing, can be proactively taken to improve software reliability. At a high level, online failure prediction approaches Operate in a similar manner. The system under observation is augmented with failure prediction models. As the augmented system runs, specific types of execution data, called system spectra, are collected and fed to the models. The models then make Predictions at runtime about whether the execution will fail or not.

Many online failure prediction approaches treat the system under observation as a black box and collect specific types of execution data that are either directly reported by the system, such as failure and error logs or directly observable from outside the system, such as CPU and memory utilization of the system. Although these approaches have been shown to be effective in predicting failures, the quality of predictions can further be improved by treating the system under observation as a white box and collecting internal execution data, i.e., by collecting data from inside executions. Collecting internal execution data, however, requires instrumenting the system under observation; the data is collected every time the instrumentation code is executed.

2. APPROACH

The lightweight online failure prediction approach, called Seer, in which most of the data collection work is performed by fast hardware performance counters. The hardware-collected data is improved with further data collected by a minimal amount of software instrumentation that is added to the systems software. In our empirical evaluations conducted on three open source projects, Seer performed significantly better than
other related approaches in predicting the manifestation of failures.

The contributions of this work can be summarized as follows:

- A novel approach for combining hardware and software instrumentation for online failure prediction and three different types of hybrid spectra produced by using this approach.
- A lightweight online failure prediction approach that uses the proposed hybrid spectra.
- A series of experiments evaluating the performance of the proposed approach and comparing it to that of other related approaches.

3. PROBLEM STATEMENT

Online failure prediction approaches aim to predict the manifestation of failures at runtime before the failures actually occur. Existing approaches generally refrain themselves from collecting internal execution data, which can further improve the prediction quality. One reason behind this general trend is the runtime overhead incurred by the measurement instruments that collect the data. Since these approaches are targeted at deployed software systems, excessive runtime overhead is generally undesirable. The large cost reductions in collecting internal execution data for online failure prediction may derive from pushing the substantial parts of the data collection work onto the hardware.

4. RELATED WORK

4.1 Applying Phases

In [1], Training phase, they are all hard coded in the prediction codes to reduce the runtime overhead. After executing the prediction code, the prediction probe applies either the point-wise or the sliding window approach (depending on the configuration). In the point-wise approach, a failure warning is issued after receiving an F prediction from a seer function. In the sliding window approach, the predictions made by seer functions are stored in an array of length l, where l is the window length being used. This array is used in a circular manner to store the last l predictions. If the sum of the scores stored in the array (i.e., window score) is greater than or equal to the window cutoff value determined in the training phase, a failure warning is issued.

The monitoring phase, violations of these invariants is flagged as errors. Fault screeners differ from each other in the types of invariants they use. In this work, distinguish between three types of fault screeners: single range screeners (SRS), multiple range screeners (MRS), and Bloom screeners (BS). Single range screeners maintain a tight range that accommodates all the values observed in the training phase. In the monitoring phase, if a value outside this range occurs, then a failure warning is issued. Multiple range screeners maintain a predetermined number of ranges, rather than a single range. When a new value is observed, the ranges are updated such that the valid range is increased by the least amount [1].

4.2 Fault Prediction Performance in Software Engineering

In [5], Fault prediction models that have the potential to improve the quality of systems and reduce the costs associated with delivering those systems most studies report insufficient contextual and methodological information to enable full understanding of a model. This makes it difficult for potential model users to select a model to match their context and few models have transferred into industrial practice. It also makes it difficult for other researchers to Meta-analyze across models to identify the influences on predictive performance. A great deal of effort has gone into models that are of limited use to either practitioners or researchers.

4.3 Count Models for Software Fault Prediction

In [10], Count Models, such as the Poisson regression model, and the negative binomial regression model, can be used to obtain software fault predictions. With the aid of such predictions, the development team can improve the quality of operational software. The zero-inflated, and hurdle count models may be more appropriate when, for a given software system, the number of modules with faults are very few. Related literature lacks quantitative guidance regarding the application of count models for software quality prediction. This study presented a comprehensive empirical investigation of eight count models in the context of software fault prediction. It includes comparative hypothesis testing, model selection,
5. PROPOSED WORK

Online failure prediction is in the forecast, while the system is running, failure occurrence in the near future, with the current state of a system, and often, the experiences from the past. Unlike other prediction methods, online failure prediction on a short-term rating on the basis (forecast for the Online near future is always more successful than Attempting long-term predictions) and prepared on the basis of the current system state (Usually by the runtime monitoring Identifies Techniques). As in noticed, reliability and resilience are and will remain in ongoing challenge due to (among others) remain the common configuration, updates and upgrades in Dynamical Systems, and the use of third-party, commercial-off-the-shelf components. For this system, runtime monitoring and online fault prediction seems to be one of the few alternatives for effective online reliability assessment and improvement.

5.1 Run-Time Verification of Dynamically Evolving Systems

A System that through specific and simple annotations enables the automatic creation of instrumented WS-BPEL processes that can be monitored. The level of monitoring can be dynamically set through a web-based interface. The MOSAICO approach to monitor (kernel-based considerable number of approaches have been developed for run-time monitoring of dynamically evolving systems. It is possible to generate a snapshot of the structure of a running application, and how this can be combined with behavioral specifications for components to check compatibility against system properties. It proposes a run-time monitoring and verification technique that can check whether dynamically adaptive software satisfies its requirements. A simple architecture) evolving component-based systems and verify interaction pattern properties. While those approaches verify dynamically evolving systems, none of them implement failure prediction features.

5.2 Failure Prediction

Online failure prediction can be most probably considered the survey analyzes and compares a number of existing online failure prediction methods. First, taxonomy is proposed in order to structure and classify the existing online failure prediction methods. Then, forty seven online prediction methods are considered and mapped into the taxonomy. Most of them use heuristics, statistics or probabilistic models to predict failures that may potentially happen in the near future. Metrics are also defined to be able to compare the failure prediction accuracy of the surveyed approaches. Although CASSANDRA belongs to the online failure prediction research branch, differently from the surveyed approaches we specialize on component based, dynamically evolving systems.

6. CONCLUSION AND FUTURE WORK

The lightweight online failure prediction approach, in which most of the data collection works is performed by fast hardware performance counters. It proposed a novel proactive monitoring approach for online failure prediction of dynamically evolving systems. The Prediction approach captures the current state of the system through proactive monitoring its execution, and uses it to explore design-time system models built on-the-fly. The approach has been initially formalized through different algorithms, and successively customized into the OSGi component framework. Some evaluations have been performed to show the feasibility of the proposed algorithms and tools on the Toast case study.
REFERENCES:


