Research Statement

Xinyu Lian

Motivation and Overview

The goal of my research is to build reliable computer systems that operate at cloudscale. Currently, my research focuses on addressing one major cause of cloud system failures – configuration errors. In large-scale cloud systems, configuration changes are deployed thousands of times on a daily basis. This velocity of configuration changes has inevitably led to configuration errors. In practice, erroneous configurations are often deployed to hundreds and thousands of nodes, leading to catastrophic outages.

My whole passion and motivation are grounded in understanding the origins of these configuration errors and developing strategies to detect and prevent them. In the first year of my Master of Science program at the University of Illinois, I have been working on three research projects studying and addressing configuration errors from different aspects.

Test Selection for Unified Regression Testing

My research started with the observation that Ctest, a promising technique for configuration testing, has the assumption that the code does not change and has only one production configuration, which is not fit to the industry practice. Therefore, Unified Regression Testing (URT) is needed to synergistically test the changed code and all changed production configurations for deployment reliability. However, URT could be very costly because it has to run a large number of tests under multiple configurations, including the default configuration and several production configurations.

We design Unified Regression Test Selection (uRTS) to effectively reduce the testing cost. uRTS supports project changes on 1) code only, 2) configurations only, and 3) both code and configurations. It selects regular tests and configuration tests with a unified selection algorithm. The uRTS algorithm analyzes code and configuration dependencies of each test across runs and across configurations. uRTS provides safety guarantee while selecting fewer tests and, more importantly, reducing the end-to-end testing time. We evaluate uRTS on hundreds of code revisions and dozens of configurations of five large projects. The results show that uRTS reduces the end-to-end testing time, on average, by 3.64X compared to executing all tests. This work was published to the 45th International Conference on Software Engineering (ICSE 2023), a premier conference on software engineering research.

Exploring Pre-Trained Large Language Models for Configuration Validation

In reflecting on my experience with the uRTS project, I've experimented with several state-of-the-art configuration validation frameworks. Unfortunately, I found that existing solutions are significantly limited. Rule-based misconfiguration detectors are challenging to maintain because rules must be manually crafted, making it difficult to scale to large, and evolving systems. However, traditional machine learning approaches require a large amount of training data, which is not always available. Moreover, traditional machine learning approaches require feature engineering, making it specific to certain types of misconfigurations.

Recent advancements in machine learning have shown that Large Language Models (LLMs) such as GPT-4 and Codex are general-purpose models without the previously described limitations. That's because LLMs can reason based on documents to determine whether the input configuration complies with them, eliminating the need for manually tailored processes. Moreover, LLMs are trained on vast corpuses of text data and are not limited to particular types.

I lead the project to undertake the exploration of employing pre-trained LLM for configuration validation, with an aim to comprehend its feasibility and effectiveness. In particular, we develop Ciri, an LLM-based configuration validation framework that integrates popular LLM models. With extensive empirical evaluation, we analyze various design choices and tradeoffs for leveraging LLMs for configuration validation. We plan to submit the work to the 46th International Conference on Software Engineering (ICSE 2024).

Cross-System Configuration Testing

While defending systems against configuration errors, I found that several configuration issues are about failures of coherently configuring multiple involved systems. This is primarily due to the nature of modern cloud systems, which are composed of independent and interacting (sub-)systems, each specializing in important services (e.g., data processing, storage, resource management, etc.). As such, cloud system reliability is affected not only by the reliability of each individual system, but also by the interplay between these systems. The recent study on these Cross-System Interaction (CSI) failures showed that 25% CSI failures are caused by configuration issues. However, few works on configuration management have touched cross-system configuration.

To address cross-system configuration failure, I lead the project to develop a novel solution, the cross-system configuration testing (CSCtest), which transforms the traditional software tests in an automated fashion that reuses well-engineered test logic and oracles to test the configuration changes in multiple systems. More precisely, we consider any software test that reads the value of a configuration parameter from another system as a potential CSCtest, that can be used to verify the correctness of cross-system configurations. Our results show that CSCtests can effectively detect failure-inducing configuration changes before deploying them to production. We reproduced 12 historical issues. By leveraging the fuzzing technique, we found several configuration-related code bugs that have not been reported before. This project is honored to be selected as the Best Research Project of CS525.

Future Research Plan

I plan to continue doing research to build reliable computer systems that operate at cloud and datacenter scale, focusing on addressing configuration errors. I have several exciting ideas in mind, such as understanding the human factor in configuration management, applying formal verification for configuration management and leveraging LLMs to automatically generate configuration testing. These research endeavors are ambitious and will require deep exploration, creativity, and a strong knowledge of cloud systems, machine learning, and software engineering. I am excited about the opportunities and challenges and look forward to making a significant contribution to the field.