Fault Localization in Concurrent System

IN4MATX 215

Milestone 2 – Project Plan

Team Member:

Chiyu Cheng

Yi Zhou

Introduction

Testing is one of the most critical and expensive tasks in the development of software. The process of detecting and locating bugs in a software system is called fault localization. Manually fault localization is acceptable for the small software system but unbearable for the large-scale system. Therefore, an automated fault localization technique is necessary for the real-world complex software system. A concurrent system is one where a computation or operation can be executed without waiting for all other operations to complete. It generally involves the coordination of multi-thread/process/cores to run a program. The concurrency fault is difficult to find because a concurrent system may have a large number of threads interleaving, which makes it is hard to reproduce a fault in one specific thread interleaving. The research question we have is, what is the best method to locate the concurrent faults in the large-scale system? There are many concurrent fault localization methods like Falcon, CCI, and etc. However, there is no comprehensive evaluation framework to evaluate and compare those techniques. An evaluation framework will guide the researchers to choose the right direction in the concurrent fault localization field. In addition, the process of evaluating different concurrent fault localization techniques will help us to understand this field better, and also help us to answer our research question

Challenges

There are many challenges to implement this evaluation framework. The biggest challenge is how to build a runnable program of each technique we evaluate. It is impossible for us to re-implement the techniques by ourselves within only one month. Therefore, we are planning to use

the existed code provided by the paper to test it. In case we are not able to find the code or the code is not runnable, we will evaluate it conceptually with depth. Another big headache is how to evaluate those techniques under the same standing since each technique has different context or definition. For the runnable techniques, we will try to run it on the SPLASH-2 program to check their performance. For those not, we will analyze it through our criteria conceptually.

Approach

In order to answer the research question, what is the best method to locate the concurrent faults in the large-scale system, we will develop an evaluation framework to study and evaluate the current state of the art in finding the concurrency fault. The plan is surveying 8-10 papers in this topic and evaluating the methods provided in those papers under the same standard.

The basic evaluation criteria are feasibility, efficiency, and accuracy. In this project, feasibility means the easiness to implement and deploy one technique to existing programs. It is hard to convince a programmer to adopt one testing technique when it is hard to implement or integrate it on their programs. Therefore, we believe that the feasibility of one technique is worth to evaluate. Efficiency includes the time complexity of the algorithm for each technique, and the resource usage like CPU, memory, or disk. Efficiency is important for most programmers since they might don't have enough time or computing resource to adopt one new technique. Accuracy stands for the success rate to find the real concurrent bugs under a certain number of executions. Without accuracy, A technique is useless even if it is feasible and efficient. For now, we are planning to use those three basic criteria to evaluate different techniques, and we might add more criteria after we finish reading more related papers.

As we motioned in the challenging part, it is impossible to implement all the techniques within one month. If we can find the runnable code for one technique, we will test if each method is feasible to integrate to our benchmark and record the performance of each method on the real system like the benchmark SPLASH-2 programs from Stanford University. For those techniques that we can't find the runnable code, we will conceptually analyze it through the criteria we motioned before with a certain level of depth.

Evaluation

The Evaluation of an evaluation framework is always tricky since it is hard to provide any data or charts. One of the most important purposes of this project is helping the author and reader to learn and understand the field of concurrent fault localization. Another essential purpose is helping researchers or programmers to find the best concurrent localization technique based on their specific requirement. Therefore, if the key reader, professor Ahmed, or the majority of the reader, agrees that this paper is helpful for achieving the two major purposes mentioned above, then this paper is qualified.