Continuous testing of concurrent software

Short Description

Continuous software testing considers execution of tests on every software change. While unit tests are perfect for this type of testing, finding concurrency bugs is much more complex. Because of implicit coupling between threads, changes in one thread affect other threads. Therefore, it is necessary to establish a more complex type of testing that includes tracing of execution and analysis of the execution trace.

This project will focus on building a docker-based environment that contains a tracer (we will be using an existing tracer tool) that can be deployed on a server. On every push of a software change to a version control (e.g., GitHub), a webhook will contact the server with the docker and send changes. Based on these changes, it will be necessary to run tests and perform analysis of the execution trace. The assumption is that unit tests are present in software. The basic, brute force approach will consider execution of all tests to generate a trace. An optimization will consider propagation of committed changes and execution of only relevant tests. For the analysis of the execution traces, we will be using basic Eraser Lockset algorithm that finds data races.