Efficient Replay for Reproducing Concurrent Program Errors

Theodorus Eric Setiadi*, Ken Nakayama,
Akihiko Ohsuga, Mamoru Maekawa

Graduate School of Information Systems, University of Electro-Communications,
Chofu-shi, Tokyo 182-8585, Japan
eric@maekawa.is.uec.ac.jp
{ken, akihiko}@ohsuga.is.uec.ac.jp
maekawa@maekawa.is.uec.ac.jp

Abstract. Debugging concurrent multithreaded programs is difficult because errors are not always reproducible. Even when the same set of input values are given, a concurrent multithreaded program might not always produce the same output as a sequential program does, because the results depend on a particular schedule and interrupt timing. When we apply tracing, the timing might be changed and the error might not occur. In other words, the exact timing that caused the error is unknown. This work proposes a method to reproduce errors, specifically race conditions, caused by different schedules and interrupt timing on a concurrent multi-threaded program. We propose a debugging/testing system that generates all possible executions as test cases based on the limited information obtained from an execution trace, and then detects potential race conditions. Our objective is to reduce redundant test cases for checking race conditions by introducing a new concept called “atomic bundle”. We show by experiment that this method significantly reduces the necessary test cases required to detect potential errors.

Keywords: concurrent multithreaded programs, error reproduction, race condition, replay system.

* Corresponding Author. The author was supported by the Jinnai international student scholarship. Email: eric@maekawa.is.uec.ac.jp.