Thesis Proposal

Sreeram Sadasivam
M.Sc Distributed Software Systems
TU Darmstadt, Germany
sreeram.sadasivam@stud.tu-darmstadt.de

July 19, 2017

Overview

Execution time and performance has always been the highlight of any multi-threading system. Migration from single processing based environment to a multi-processing environment has led to the creation of various synchronization problems. Some of the key problems being race condition, deadlocks, maintaining execution order, etc,. One possibility is to comprehend various thread execution scenarios inorder to neutralize the above mentioned problems. Deterministic threading or sometimes referred to as DThreads is one such scheme indicated to resolve the above possibility. There have been various frameworks and publications comprehending the possibility to use determinism in regard with the user's program. All these frameworks have fell short in improving execution time, easiness and robustness in integrating with the user's existing code base, etc,. Most of these problems are correlated in having a use of a proper thread scheduler and also having a well-designed abstraction for the user.

In the proposed design, we have thread scheduler and an instrumentation framework. The instrumentation framework primarily focusses on instrumenting the user code and comprehending memory accesses made by various user threads. The expectation is for the user to specify valid thread scheduling patterns into the framework along with his/her program. The framework

generates a safe threading scheduling pattern. The generated execution pattern is later realized with the thread scheduler, when the user program is executed.

The design can be realized by having the thread scheduler as part of the framework thus, making the implementation easier but having the thread decisions made in the user land. Or, we can package the thread scheduler as a Loadable Kernel Module. Thus, isolating the scheduler from user land.

The design would be benchmarked on various thread conditional scenarios such reader-writer problems, Peterson solution, Lamport solution.