Thesis Proposal

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

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 point of correctness in result. Deterministic multi-threading is one such scheme indicated to resolve the above possibility. There have been research insights into the area of fine-grained scheduling of multi-threaded programs. But, most of the solutions have been reliant on the run-time systems's load balance engine. Thus, leaving the possibility to perform a verification of the execution program before run-time. Most of the existing solutions have been dealt with various compactibility issues.

In the proposed design, we have thread scheduler and an verification engine. The verification engine primarily focusses on instrumenting the user code and realizing memory accesses made by various user threads. The set of safe schedules are provided by the verification engine for the given user program. 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 a Loadable Kernel Module. Thus, isolating the scheduler from user land. And also reducing significant execution time overhead.

The design would be benchmarked on various thread conditional scenarios such reader-writer problems, Peterson solution, Lamport solution. These programs enforce the verification of correctness in multi-threaded environment. The evaluation is performed on the overhead exerted by the transition to the LKM module. And also comparing the likes of the rival designs PARROT, CORE-DET. The evaluation is scaled from the use of thread count pertaining to the core count, to the use of scaled up version of thread count overshadowing the core count. Thus, creating a possibility of false sharing situations and various other potential execution overhead conditions.