

Summary

Model-Based Schedulability Analysis of Hard Real-Time Java Programs using Software Transactional Memory

Group sw103f12

Marcus Calverley
Anders Christian Sørensen

This report documents our work in developing a software transactional memory (STM) for real-time Java, which assigns priorities to transactions based on the tasks in which they execute. Contention is managed in such a way that the highest priority transaction does not experience retries, allowing for irreversible actions such as I/O, which is otherwise impossible using traditional STM. These properties are proven using the model checking tool UPPAAL.

The Hardware-near Virtual Machine (HVM) is chosen as the target platform due to its portability and flexibility, but it does not support multi-threading in its current state. To alleviate this, we have implemented real-time multi-threading through an implementation of Safety-Critical Java named SCJ2.

In order to determine schedulability of real-time Java programs using our STM and SCJ2, we have developed \systemname, which is a model-based schedulability analysis tool. It analyses the Java bytecode of the input program, and produces a UPPAAL model which can be used to verify schedulability. We have conducted experiments with our tool, comparing it to a similar existing model-based schedulability analysis tool named SARTS. We also compare the use of locks and our STM in a real-time setting, showing their advantages and disadvantages.