

Master's Thesis

« Interoperable Benchmark Framework for Transactional Memory »

The move to multi-core architectures in the end user domain currently happening forces the software industry to move to parallel programming techniques. Parallelisation is known to be a complex task and still challenging for experienced programmers as well. Thus, many researchers currently focus on utilities, methods and concepts to ease this task. Transactional Memory (TM) provides a framework to apply transactions on data structures in memory. Transactions are deadlock-free and provide better scalability as traditionally used locks. Thus, it is a promising tool for the future. TM implementations are available as hardware extensions, software libraries, as hybrids of both types and even the concurrency control algorithms used in TM are very diverse. Thus, it is hard to evaluate certain TM implementation against others. Available benchmarks cover just a given set of usage scenarios. To evaluate certain usage scenarios not covered by them one (1) has to implement a new benchmark and (2) attach the TM-implementations to it, he'd like to compare. Recently some researchers try to fill this gap with some kind of interoperable benchmark framework. It's based on a C/C++ compiler, which understands keywords used to mark critical regions as part of a transaction. On the other hand, the same compiler provides an interface to compiler backends which produce suitable code for certain TM runtime libraries. Thus, a benchmark, written for the compiler can run on all TM implementations supported by the compiler. The good news is, there are now a few compilers for TM understanding TM specific keywords and providing an interface for TM backends. The bad news is, those compilers use similar but not the same specifications for their interfaces and the benchmarks and backends run just with a specific compiler. Thus, the problem has just been moved to another level of abstraction and is not solved! So, the aim of the master thesis provided here is to review and define interoperability for TM compilers and benchmarks in way of an interoperable benchmark suite.

Tasks

- Analysis of the disadvantages of current TM benchmark suites and existing attempts for interoperability
- Define interoperability in respect to TM benchmarks and design an interoperable benchmark framework
- Implementation of a prototype of the interoperable benchmark framework
- Evaluation of the prototype
- Documentation according to research standards (master thesis text)

Required Knowledge

- C/C++ development under linux

Contact: Holger Machens

machens@tu-harburg.de

Phone: +49 40 / 428 78 – 3703

Room: E 4.090