

# Master's Thesis

## « A Runtime Environment for Thread Level Speculation »

The multi-core architectures available today provide real parallel processing power in the end user domain. To leverage the constantly increasing number of cores the software industry is forced to use parallelisation techniques. One such parallelisation technique is thread level speculation (TLS) where a thread can employ one or more helper threads (so-called shadows) to speculatively pre-execute future code sections while itself proceeds through the code at the current position. The result of pre-executed code sections may be used by the parent thread later if the predicted branch is actually taken and thus effectively reduces the processing time for the given thread. This techniques are almost similar to those known from pipe-lining in CPUs combined with branch prediction. Task of this bachelor thesis is to design and implement a runtime environment to provide TLS capabilities on application level. It is thought to be based on a specific thread model where each thread and shadow is represented by a process which allows to copy the thread context (stack, registers, interrupt mask) between threads without invalidating pointers in the stack. In this thread model shared data is located in a shared memory segment which also allows the thread to have efficient access to thread local storage. The focus lies on the design of the threading model with an appropriate API to delegate work to shadows, memory management on top of shared memory, a system-wide resource management to assign shadows to cores as well as appropriate synchronization capabilities such as spin-locks, spin-lock-based barriers and so on. Low-level functions to copy the thread context, a simple memory management and a simple locking function is already available.

### Tasks

- Analysis of methods to delegate work for TLS and reviewing of existing APIs for threading, memory management, locking etc.
- Specifying the TLS runtime environment and its APIs.
- Implementation of the TLS runtime environment (at least its core functionality)
- Evaluation by realisation of simple TLS methods.
- Documentation according to research standards (bachelor thesis text)

### Required Knowledge

- C/C++ development under Linux

**Contact: Holger Machens**

[machens@tu-harburg.de](mailto:machens@tu-harburg.de)

Phone: +49 40 / 428 78 – 3703

Room: E 4.090