

Bachelor's Thesis, Master's Thesis

Tracing of thread execution and memory allocation

The scheduling of different threads on the CPUs and the assignment of the corresponding data into distributed memory regions highly influence the timing behaviour of parallized applications. In order to measure the impact and evaluate scheduling stratagies we would like to have a non-intrusive library for measuring the executed actions and influencing the corresponding scheduling of test programs and benchmarks.

Goal

Your major goal will be to develop a library which replaces the corresponding function calls to the C library with an implementation keeping track of the requested actions. The main focus will be on functions for thread creation, interruption (e.g. semaphores and mutex), and memory allocation and deallocation. The collected data must be stored into log files without highly influencing the timing behaviour of the test program. Furthermore the functions should provide a configuration interface in order to influence the scheduling and data placement.

Finally an evaluation of the collected traces based on statistical data analysis will be part of the work.

Prerequisites

To successfully complete this project, you should already have the following skills and experiences:

- Solid C/C++ programming skills
- Good knowledge about multiprocessor architectures
- Basic knowledge about operating systems and thread APIs
- Self-motivated and structured work style

Contact

Dirk Gabriel
Room N2117
Tel. 089 289 28578
dirk.gabriel@tum.de

Advisors

Dirk Gabriel