Master’s Thesis

« Linux Kernel Enhancement for Efficient Threading and Thread Level Speculation Capabilities on Multi-Core Architectures »

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 master thesis is to design and implement a runtime environment to provide TLS capabilities on application level. The core task is to develop a Linux Kernel patch which allows efficiently copying/exchanging the thread context (stack, registers, interrupt mask) between threads without invalidating pointers in the stack. This includes the design of an enhanced memory management model for linux where each thread maps its stack to the same virtual address (but a different physical address). This enhanced memory model also allows a very efficent access to thread local storage which is also benefitial for TLS and concurrent programming in general. Another part of the master thesis is the design of an appropriate API to delegate work to shadows and a system-wide resource management to assign shadows to cores and share shadows between multiple processes. On success the results of this thesis will be published as a kernel patch and proposed as enhancement to future Linux Kernel releases.

Tasks

- Analysis of the Linux Kernel to insert required extensions.
- Analysis of methods to delegate work to shadows and manage shadows system-wide.
- Specifying the TLS runtime environment, Kernel modifications and its APIs.
- Implementation of the TLS runtime environment (at least its core functionality)
- Evaluation by realisation of meaningful TLS test cases.
- Documentation according to research standards (master thesis text)

Required Knowledge

- C(++) development under Linux
- The ability to read and understand C source code
- Advantageous but not required: Knowledge in Linux Kernel development

Contact: Holger Machens

machens@tu-harburg.de Phone: +49 40 / 428 78 – 3703 Room: E 4.090