Project LAOS: Latency Awareness in Operating Systems *

Gabor Drescher

Friedrich-Alexander Universität Erlangen-Nürnberg
drescher@cs.fau.de

The goal of the LAOS project is to investigate in the efficient use of modern many-core processors on operating system level. Thereby providing low latency operating system services even in high contention cases. Self-made minimal kernels providing thread and interrupt management, as well as synchronization primitives are analyzed with respect to performance and scaling characteristics. These kernels consist of different architectural designs and alternative implementations. Strong focus lies on non-blocking implementations on parts, or if possible, on the whole operating system kernel. Standard Intel x86-64 compatible processors are the main target hardware because of their popularity in high-performance parallel computing, server and desktop systems.

Two operating system kernels were implemented, they serve twofold purposes. First, analyzing the effects of different kernel architectures on the overall performance of the system. And second, analyzing the possibility of non-blocking synchronization in kernels and its effects on system performance. One kernel has a multicore event-based architecture that, namely, provides only a single kernel stack for all threads sharing the same processor core. Whereas the second kernel is process-based, which implies the allocation of kernel stacks on a per-thread basis. Both kernels are non-blocking, meaning that all concurrent operations on data structures are carried out transactionally via atomic instructions of the CPU, no spinlocks where used. Both kernels provide an interface for the creation, management and synchronization of threads in a shared memory context.

Custom numerical benchmarks were implemented, comparing Linux and the LAOS kernels on the same 48 core cache-coherent NUMA machine. Speedups of 25% over the performace of Linux could be achived, as well as much better scaling in high contention scenarios with execution times 4x faster than Linux.

* This work was supported by the German Research Foundation (DFG), partly under grant no. SCHR 603/8-1 and by the Transregional Collaborative Research Centre "Invasive Computing" (SFB/TR 89).