論文の内容の要旨

Operating System Organization for Manycore Systems
（メニーコアシステムにおけるオペレーティングシステムの構成）

氏　名　下沢　拓

Manycore processors, which have tens or hundreds of cores, have emerged to gain computing capacities by increasing parallel computing capacities. There are two possible forms of utilization of manycore processors in computing clusters: one is that manycore processors work as the main CPU in a node, and the other is that they are employed as accelerator in addition to the host multicore processors. The former style is considered to be a natural extension of the current multicore clusters, where the number of the cores increases largely. It is assumed that all the memory and processors are connected with lower-latency inner interconnects in this case. The latter has multicore host CPUs and manycore accelerators connected with each other by generic I/O buses. The host CPUs have relatively higher single-core performance and larger cache and faster memory buses compared to manycore processors.

There are three issues to design operating systems for such systems. One is the scalability of the operating system kernel when the number of the processor cores increases, which is important especially in the former case, where the manycore processors must manage the operating system services and I/O processing by their own. Another is the cache pollution caused by the operating system kernel since the manycore processors are relatively sensitive to cache misses compared to multicore processors. The other is only for the latter case; the memory spaces of host CPUs and accelerators are separate and that it is difficult to run existing operating systems, which assume cache-coherent shared memory.

In this thesis, we propose two forms of operating systems for manycore systems to mitigate these issues. First, SHIMOS (Single Hardware with Independent Multiple Operating Systems) is proposed for use in the former system. It partitions the machine by multiple kernels to reduce the number of processors running to alleviate contentions. It has zero-copy communication mechanism between the kernels to allow all the kernels to access the devices. Second, HIDOS(Host Inter-kernel Delegation Operating System) is presented for use in the accelerator system. It executes a commodity operating system kernel in the host processors and small kernels in the accelerator processors to alleviate cache pollution by delegating the system services on accelerators to the kernel in the host. It is based on several abstraction layers to enable development of operating systems without any target accelerators, to serve one-copy inter-kernel communication, and to realize the delegation mechanism on it.

The design and implementation of SHIMOS and HIDOS on Linux in x86 commodity servers are presented in this thesis. SHIMOS is evaluated to demonstrate that it achieves partitioning with almost no overhead and runs I/O-loaded benchmarks faster than a single Linux by reducing contentions. HIDOS is also evaluated in the emulated environment on the commodity server and shows that it reduces cache pollutions and performs better in parallel applications with system calls compared to a single Linux.