

III.6. "Protocol Processors vs. Custom Hardware Adaptors for SMP-based
CC-NUMA Multiprocessor Architectures"(R)
by Maged M. Michael, Beng-Hong Lim, Ashwini K. Nanda, Michael L. Scott
University of Rochester and IBM T.J. Watson Research Center

A recent trend in building large-scale cache coherent multiprocessors is to use commodity SMP nodes as building blocks and a custom-designed coherence adaptor to extend cache-coherent shared memory across the nodes (e.g. the Sun S3.mp [ICPP 1995] and the Sequent STiNG [ISCA 1996]). The coherence adaptor is the most complex custom-designed part in the development cycle of such CC-NUMA systems, and a key design issue is whether to implement the coherence protocol entirely in hardware on the adaptor, or partially in software using a protocol processor on the adaptor. The former approach promises better performance while the latter approach allows more flexibility and shorter development time.

Any well-designed computer system needs to strike a careful balance among the various components of the system, and the design of a coherence adaptor is no exception. The tradeoff between custom hardware and a protocol processor depends on other system components such as network speed, the presence of a remote access cache, and cache directory access latency. We investigate the design of such adaptors under various architectural configurations of SMP-based CC-NUMA multiprocessors and application workloads. We determine where the performance of the protocol processor approach is sufficient and where the additional performance from an all-hardware implementation is necessary.

When a slow network is used, remote memory access latency is dominated by the network delays, and so the additional penalty of a protocol processor may not have a large impact on memory system performance. However, if the network is fast, the penalty of a protocol processor may be relatively high. A remote access cache reduces SMP-to-SMP interaction, thus reducing the overall performance penalty of protocol processors. Directory access latency has complex effects on both the local SMP bus traffic and the speed of the cache coherence protocol. Slow access to directory state might require extending the response window for ALL bus traffic and decrease the possible rate of SMP-to-SMP interaction, thus decreasing the performance penalty of protocol processors. However, slow access to directory state would slow down protocol processing, thus increasing the performance penalty of protocol processors. Because these influences conflict, it is unclear whether the net effect of slow directory access argues for or against protocol processors.

To study these effects, we have developed an execution-driven simulation environment (using the Augmint toolkit) for Intel x86 as well as IBM RS/6000 based multiprocessor architectures. The simulation environment includes models for processor caches, Intel Pentium Pro and IBM 6XX buses with MESI cache coherence protocols, memory controllers, coherence adaptors, and a directory-based cache coherence protocol. We are in the process of modeling remote access caches and interconnection networks. We use applications from the SPLASH-2 suite of parallel applications with various memory access characteristics for performance evaluation of the alternative approaches.

Heinrich et al. [ASPLOS-VI 1994] studied the performance impact of using protocol processors in the FLASH multiprocessor with uniprocessor nodes. In contrast, we study a CC-NUMA multiprocessor with SMP nodes. Also, their work evaluates the performance penalty of protocol processors on a specific design point, FLASH, while our work considers the effect of multiple architectural components, network delay, remote access caches, and directory access latency on the performance penalty.