ABSTRACT

Efficient CPU-GPU Communication for Heterogeneous Architectures

MOTIVATION: Future chip multiprocessors (CMPs) will have silicon space and technology to incorporate hundreds of cores. The trend is to integrate tens of cores and hardware accelerators (HAs), such as GPUs, on a single platform. This heterogeneous architecture will enable future chips to operate within their power budgets while providing the high-throughput per Watt required for large scientific applications. Many of the top-500 supercomputers integrate thousands of CPUs with GPU accelerators to achieve the desired throughput for scientific applications. Considerable effort, however, is needed to design efficient communication mechanisms between heterogeneous components in such a system. Currently, HAs are not fully integrated with the system architecture; offloading computation from the CPU to the HAs adds large communication overhead. This research project explores comprehensive solutions to this problem through many different techniques. The project has significant broader impact in terms of research publications, graduate student supervision, and minority education because UCR is a minority serving institution.

TECHNICAL DESCRIPTION: This project develops new CPU-GPU communication techniques through static programming and run-time optimization. It develops a divisible load theory (DLT) technique to overlap communication with computation, and optimizes the time and size of data transfer between the CPU and GPU. The research also develops run-time techniques that can monitor the efficiency of execution and dynamically change the transfer parameters by considering the execution behaviors of different applications. Architectural changes are incorporated in the GPU to initiate data transfers based on task execution inside the GPU.

Design of the shared virtual memory (SVM) architecture is developed, where the accelerator and system memories share a single virtual address space; CPUs and HAs in the system communicate through the SVM. The hardware controllers, memory management unit (MMU), GPU cache memory architectures, cache coherence protocols, and other interfaces between the GPU and CPU cores are also designed. The project proposes suitable hybrid cache coherence protocols and efficient interconnection networks for scalable system design. Finally, run-time system and software interfaces are developed that can execute multiple multithreaded applications on a heterogeneous multicore architecture.