Accelerating the Execution of Matrix Languages with the Cell Broadband Engine Architecture

Raymes Khoury
rkho9564@it.usyd.edu.au

Bernhard Scholz
scholz@it.usyd.edu.au

Bernd Burgstaller
bburg@cs.yonsei.ac.kr

Abstract

High-level matrix-based languages such as MATLAB, Octave and Modellica are established standards for rapid-prototyping in scientific and engineering domains. As the processing speed of single-core processors levels out due to clock speed limitations, modern computer architectures have been introducing parallel high-speed execution units, which need to be fully utilised to obtain peak-performance. However, high-level matrix-based languages have underlying sequential execution semantics and do not support an explicit notion of parallelism.

We present a new framework in the form of a non-intrusive Octave extension which is designed to exploit the computational power of modern parallel architectures without placing additional burdon on the programmer. Lazy evaluation is used to delay the immediate execution of matrix operations and a data-dependence graph of these operations is constructed. When an uncomputed result is required, these operations are partitioned and scheduled among the parallel units of the system before being executed. In this way we facilitate parallelism of matrix calculations in two ways: (1) by executing independent operations in parallel, and (2) by dividing large operations into smaller ones which are executed in parallel. By estimating the execution time of operations and scheduling them among parallel units, better utilisation of the architecture can be obtained.

Our first implementation of the framework is for the Cell processor, however the framework is easily extendable to other parallel architectures such as General Purpose Graphics Processing Units (GPGPUs) and multi-core CPUs. Our implementation uses features of the Cell processor including double buffering and SIMD operations to obtain good performance.

Importantly, we provide a model of the pipelined execution of matrix operations on the Cell processor, which is used to describe the architecture-specific scheduling problem. A new ILP formulation of the scheduling problem is described, as well as a simple heuristic algorithm and the performance of the solutions produced by each technique is compared.

Early results show that our system achieves speedups over MATLAB code running on recent Intel Core2 Quad processors.