Beyond Gaming: Programming the PLAYSTATION®3 Cell Architecture for Cost-Effective Parallel Processing

Rodric Rabbah
IBM Watson Research Center
Hawthorne, NY

ABSTRACT

The Cell Broadband Engine (BE) architecture is a 9-core heterogeneous processor [3]. It is a new and versatile architecture that is well suited for a variety of applications including digital media, entertainment, communications, medical imaging, security and surveillance, and HPC workloads. The same Cell BE processor that powers IBM Cell blade servers is also available for a fraction of the cost in Sony PLAYSTATION3 (PS3) gaming consoles. The PS3 is not just a gaming console however because it is readily possible to install other operating systems and boot the PS3 into a programmable environment using popular Linux distributions. Thus a PS3 provides a practical vehicle for academic and research endeavors that focus on parallel architectures and parallel programming. This tutorial demonstrates the ease of leveraging PS3 consoles as low-cost and high-performance platforms for parallel execution. The tutorial provides a brief technical overview of the Cell architecture, and focuses on programming models and programming patterns that facilitate the development of efficient applications for the Cell BE.

Categories and Subject Descriptors

D.1 [**Software**]: Programming Techniques—*Concurrent Programming, Parallel Programming*

General Terms

Languages, Performance

Keywords

Cell, PLAYSTATION3, StreamIt, Streaming

1. OVERVIEW

The era of performance gains for sequential programs due to technology scaling is over. Now and increasingly in the future, the primary method of increasing program performance will require the utilization of multicore parallelism.

The PS3 provides a convenient platform for experimenting with programming models and programming patterns for concurrent and parallel execution. The Cell BE differs from existing multicore offerings from Intel and AMD in that eight of its cores, which are known as synergistic processing elements or SPEs, have to explicitly fetch data from main memory into their local storage to carry out meaningful computation. Thus programs that run on the Cell BE require the orchestration of computation and communication, and therefore present many challenges that are worth exploring for research purposes. For example, efficient execution on the Cell BE requires partitioning of code across SPEs to increase parallelism, and since each SPE is a short vector processor, the code also requires vectorization for maximum efficiency. There is also the issue of synchronization between cores, and overlapping communication and computation to maximize utilization. In addition there is a large design space to explore in terms of static versus dynamic scheduling of computation, as well as space and time multiplexing of computation on the cores. The PS3 offers a common platform for innovation and experimentation, and new parallel programming solutions that work well on architectures like the Cell processor are likely to make a lasting impact on future multicore architectures as well.

The Cell BE also provides students with hands on experience to learn about the many difficulties in programming for concurrency and parallelism. The emerging trend of packing more cores on a chip highlights the need to train students to think about parallelism so that parallel programming is as easy and tractable as sequential programming. The Cell BE is a practical vehicle to catalyze this educational process. Students with good understanding of conventional programming practices can readily learn Cell programming prerequisites and begin the process of developing non-trivial applications that achieve robust and scalable performance [1].

The tutorial highlights the main challenges in programming the Cell BE. It also introduces a convenient stream programming approach based on the StreamIt [2] language to facilitate the mapping of computation to the Cell architecture. Stream programming affords a simple methodology for achieving robust and scalable performance on multicores. The tutorial demonstrates how the StreamIt compiler and runtime system provide a rich environment for rapid application development, as well as provide a framework for research in compiler optimizations and scheduling heuristics for efficient multicore execution.

CODES+ISSS'07, September 30–October 3, 2007, Salzburg, Austria.
ACM 978-1-59593-824-4/07/0009.

References

[1] Multicore programming primer. `http://cag.csail.mit.edu/ps3`.

[2] StreamIt homepage. `http://cag.csail.mit.edu/streamit`.

[3] H. P. Hofstee. Power Efficient Processor Architecture and The Cell Processor. *HPCA*, 00:258–262, 2005.