Applications Session

Cellflow: a Parallel Application Development Environment with RunTime Support for the Cell BE Processor

M. Ruggiero, M. Lombardi, M. Milano and L. Benini
University of Bologna - DEIS, Bologna, Italy,
{martino.ruggiero, michele.lombardi2, michela.milano, luca.benini}@unibo.it

Abstract

Cell is a heterogeneous multi-core architecture composed by a standard general purpose microprocessor (called PPE), with eight co-processing units (called SPEs) integrated on the same chip. The SPE is a processor designed for streaming workloads, featuring a local memory, and a globally-coherent DMA engine. Cell has already demonstrated impressive performance ratings in computationally intensive applications and kernels mainly thanks to its innovative architectural features. The heterogeneity of its computational capability, the limited, explicitly-managed on-chip memory and the multiple options for exploiting hardware parallelism, make efficient application design and implementation a major challenge. Efficiently programming requires to explicitly managing the resources available to each SPE, as well the allocation and scheduling of activities on them, the storage resources, the movement of data and synchronizations, etc.

Moving from these considerations, the novelty of this work is the creation of a framework, called Cellflow, which can help programmers in handling these complex and critical activities and decisions. Our goal is to enable developers to quickly build multi-task applications using an explicit parallel programming model. Our key object is to give developers access to the power of Cell multi-core architecture, but at a high level. We want to set programmers free from the issue of managing allocation and scheduling tasks, so they can focus on developing the core algorithms of the application.

Our toolkit is made of an off-line and an on-line component. The off-line facility is a design-time software optimization infrastructure for the deployment of multi-task applications. It is made up of:

- a generic customizable application template that helps software developers to easily and quickly build their application skeleton starting from a high level task and data flow graph,
- an allocation and scheduling support, featuring both a heuristic suboptimal algorithm and an exact solver that finds the optimal mapping and scheduling on the hardware architecture.

The software framework is targeted towards statically-configured applications, where optimal allocation and scheduling settings are pre-computed at design time. As far as the mapping and scheduling support is concerned, we have implemented two approaches: the first is based on traditional list scheduling and round robin allocation, which are simple and scalable heuristics that do not provide any optimality guarantee. The second is an exact optimization algorithm providing optimal solutions. It has been extensively described in and it is based on a multi-stage decomposition. An extensive set of experimental results confirm that the multi-stage decomposition pays off in terms of efficiency w.r.t. a traditional approach.

The on-line runtime support is composed by software libraries and APIs which manage coordination issues, such as task allocation and scheduling, as well as task-level issues, like inter-task communication and synchronization. In this phase we tackle also the problem of the limited memory size of the SPEs, optimizing their utilization through overlaying.
With Cellflow, Cell programming becomes simpler, but at the same time it achieves high efficiency thanks to the run-time support (which is tuned to the SPE hardware) and to off-line optimal allocation and scheduling.