1 INTRODUCTION - MOTIVATION

- Technology advances have enabled the development of a plethora of computation substrates for parallel, high performance computing.
- A paradigm shift towards parallel programming.
- A multitude of programming models have been proposed that aim at handling most of the technicalities, thus allowing the programmer to focus on algorithmic issues. However, none of them target both homogeneous and heterogeneous multicores as well as accelerators.

OpenCL presents itself as a unified programming standard and framework for heterogeneous multicores that integrate both CPUs and accelerators, such as the Cell BE or GPUs. Although, currently available, vendor specific implementations target either only multicores or accelerators, due to their vast architectural differences.

Motivation: Design and develop GLOpenCL, a unified framework to enable native execution of OpenCL applications on both hardware- as well as software-controlled cache multicores. The framework comprises a compiler and a run-time library that shares the same basic architecture across all platforms.

2 OPENCL PROGRAMMING MODEL

- OpenCL models the underlying parallel architecture as a host and a number of compute devices. Each compute device integrates a number of compute units, each one divided into processing elements.

- Efficient exploitation and mapping of the available parallelism is not a trivial task.

3 COMPILER SUPPORT

- Source-to-source transformations that aim at coarsening the granularity of a kernel function from a per-logical-thread to a per-workgroup basis. These transformations enable an efficient mapping of the available parallelism on multicores without hardware support for fine-grained threading.

- Logical thread serialization has the additional implicit effect of potentially reducing the local storage that is required by each work-item.

4 RUN-TIME SUPPORT

- Evaluate the framework’s performance using a series of six benchmark applications on the Intel E5520 i7 and the Cell BE processors, and compare with the SDKs provided by ATI/AMD and IBM.

5 PERFORMANCE EVALUATION

- Evaluate the framework’s performance using a series of six benchmark applications on the Intel E5520 i7 and the Cell BE processors, and compare with the SDKs provided by ATI/AMD and IBM.