Abstract

High performance computing (HPC) is undergoing a period of enormous change. Due to the difficulties in increasing clock frequency indefinitely (i.e., the breakdown of Dennard's scaling and power wall), the current direction is towards improving performance through increasing parallelism. However, there is no clear consensus yet on the best architecture for HPC, and different solutions are currently employed. As a consequence, applications targeting a given architecture can not be easily adapted to run on alternative solutions, since this would require a great effort due to the need to deal with platform-specific details. Since it is not known *a priori* which HPC architecture will prevail, the Scientific Community is looking for a solution that could tackle the above mentioned issue. A possible solution consists in the adoption of a high-level abstraction development strategy based on Domain Specific Languages (DSLs). Among them, OpenCAL (Open Computing Abstraction Layer) and OPS (Oxford Parallel Structured) have been proposed as domain specific C/C++ data parallel libraries for structured grids. The aim of these libraries is to provide an abstract computing model able to hide any parallelization detail by targeting, at the same time, different current (and possibly future) parallel architectures. In this Thesis, I have contributed to the design and development of both the OpenCAL and OPS projects. In particular, my contribution to OpenCAL has regarded the development of the single-GPU and multi-GPU/multi-node components, namely OpenCAL-CL and OpenCAL-CLM, while my contribution to OPS has regarded the introduction of the OpenMP 4.0/4.5 support, as an alternative to OpenCL, CUDA and OpenACC, for exploiting modern many-core computing systems. Both the improved DSLs have been tested on different benchmarks, among which a fractal set generator, a graphics filter routine, and three different fluid-flows applications, with more than satisfying results. In particular, OpenCAL was able to efficiently scale over larger computational domains with respect to its original implementation, thanks to the new multi-GPU/multi-node capabilities, while OPS was able to reach near optimal performance using the high-level OpenMP 4.0/4.5 specifications on many-core accelerators with respect to the alternative low-level CUDA-based version.