Abstract

OKL: A Unified Language for Parallel Architectures

David Medina

Rapid evolution of computer processor architectures has spawned multiple programming languages and standards. This thesis strives to address the challenges caused by fast and cyclical changes in programming models. The novel contribution of this thesis is the introduction of an abstract unified framework which addresses portability and performance for programming manycore devices. To test this concept, I developed a specific implementation of this framework called OCCA. OCCA provides evidence that it is possible to achieve high performance across multiple platforms.

The programming model investigated in this thesis abstracts a hierarchical representation of modern manycore devices. The model at its lowest level adopts native programming languages for these manycore devices, including serial code, OpenMP, OpenCL, NVIDIA's CUDA, and Intel's COI. At its highest level, the ultimate goal is a high level language that is agnostic about the underlying architecture. I developed a multiply layered approach to bridge the gap between expert "close to the metal" low-level programming and novice-level programming. Each layer requires varying degrees of programmer intervention to access low-level features in device architectures.

I begin by introducing an approach for encapsulating programming language features, delivering a single intermediate representation (OCCA IR). Built above the OCCA
IR are two kernel languages extending the prominent programming languages C and Fortran, the OCCA kernel language (OKL) and the OCCA Fortran language (OFL). Additionally, I contribute two automated approaches for facilitating data movement and automating translations from serial code to OKL kernels.

To validate OCCA as a unified framework implementation, I compare performance results across a variety of applications and benchmarks. A spectrum of applications have been ported to utilize OCCA, showing no performance loss compared to their native programming language counterparts. In addition, a majority of the discussed applications show comparable results with a single OCCA kernel.