Autotuning OpenCL Workgroup Sizes

Chris Cummins, University of Edinburgh

May 17, 2018

The physical limitations of microprocessor design have forced the industry towards increasingly heterogeneous designs to extract performance, with an an increasing pressure to offload traditionally CPU based workloads to the GPU. This trend has not been matched with adequate software tools; the popular languages OpenCL and CUDA provide a very low level model with little abstraction above the hardware. Programming at this level requires expert knowledge of both the domain and the target hardware, and achieving performance requires laborious hand tuning of each program. This has led to a growing disparity between the availability of parallelism in modern hardware, and the ability for application developers to exploit it.

The goal of this work is to bring the performance of hand tuned heterogeneous code to high level programming, by incorporating autotuning into *Algorithmic Skeletons*. Algorithmic Skeletons simplify parallel programming by providing reusable, high-level, patterns of computation. However, achieving performant skeleton implementations is a difficult task; skeleton authors must attempt to anticipate and tune for a wide range of architectures and use cases. This results in implementations that target the general case and cannot provide the performance advantages that are gained from tuning low level optimization parameters for individual programs and architectures. Autotuning combined with machine learning offers promising performance benefits by tailoring parameter values to individual cases, but the high cost of training and the ad-hoc nature of autotuning tools limits the practicality of autotuning for real world programming. We believe that performing autotuning at the level of the skeleton library can overcome these issues.

In this work, we present *OmniTune* — an extensible and distributed framework for autotuning optimization parameters in algorithmic skeletons at runtime. OmniTune enables a collaborative approach to performance tuning, in which machine learning training data is shared across a network of cooperating systems, amortizing the cost of exploring the optimization space. We demonstrate the practicality of OmniTune by autotuning the OpenCL workgroup size of stencil skeletons in SkelCL. SkelCL is an Algorithmic Skeleton framework which abstracts the complexities of OpenCL programming, exposing a set of data parallel skeletons for high level heterogeneous programming in C++. Selecting an appropriate OpenCL workgroup size is critical for the performance of programs, and requires knowledge of the underlying hardware, the data being operated on, and the program implementation. Our autotuning approach employs the novel application of linear regressors for classification of workgroup size, extracting 102 features at runtime describing the program, device, and dataset, and predicting optimal workgroup sizes based on training data collected using synthetically generated stencil benchmarks.

In an empirical study of 429 combinations of programs, architectures, and datasets, we find that OmniTune provides a median 3.79× speedup over the best possible fixed workgroup size, achieving 94% of the maximum performance. Our results demonstrate that autotuning at the skeletal level — when combined with sophisticated machine learning techniques — can raise the performance above that of human experts, without requiring any effort from the user. By introducing OmniTune and demonstrating its practical utility, we hope to contribute to the increasing uptake of autotuning techniques into tools and languages for high level programming of heterogeneous systems.

Autotuning OpenCL Workgroup Sizes

Tuning GPU Stencils with machine learning outperforms human experts

3.79x speedup!

Predicting OpenCL workgroup sizes of 429 stencil programs, execution devices, and datasets.

Chris Cummins Pavlos Petoumenos Michel Steuwer Hugh Leather

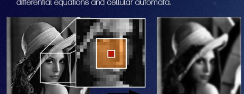
c.cummins@ed.ac.uk

Hand tuning programs is expensive and time consuming

We automate this tuning using collaborative machine learning

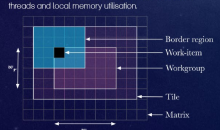
Stencil Skeletons

Stencil Skeletons are a common data parallel pattern with a range of applications from image processing to partial differential equations and cellular automata.



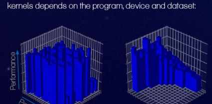
OpenCL Workgroup Size

OpenCL stencil skeletons are parameterised with a workgroup size, which controls grouping of hardware threads and local memory utilisation.

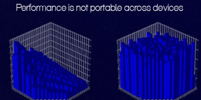


Optimization Space

Choosing the right OpenCL workgroup size for stencil kernels depends on the program, device and dataset:



Performance is not portable across devices



Performance is not portable across programs

Implemented using SkelCL skeleon library.

Introducing OmniTune ...

- OmniTune generates synthetic benchmark programs to use for empirical testing
- OmniTune collaboratively gathers performance data by testing different parameter values
- OmniTune uses machine learning to predict parameters for unseen programs at runtime

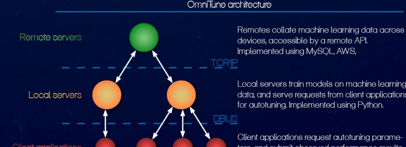
Machine Learning features

Program – LLVM bytecode
- Instruction densities
- Num. basic blocks
- Stencil region size
- etc.

Device – OpenCL API
- Compute devices
- Memory sizes
- Cache types
- etc.

Dataset – SkelCL container
- Container width
- Container heigh
- Input data type
- etc.

OmniTune architecture



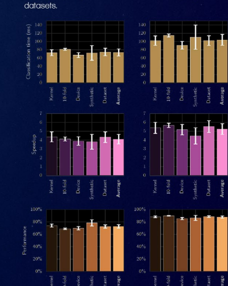
Remotes collate machine learning data across devices, accessible by a remote API. Implemented using MySQL, AWS,

Local servers train models on machine learning data, and serve requests from client applications for autotuning. Implemented using Python.

Client applications request autotuning parameters, and submit observed performance results. Implemented using SkelCL (C++ & OpenCL).

Results

Evaluated using 429 combinations of programs, devices, and datasets. Machine learning classification using linear regressors to predict program runtime of program speedup. Prediction quality evaluated across devices, programs, and datasets.



Read more ...

C. Cummins, P. Petoumenos, M. Steuwer, H. Leather "Autotuning OpenCL Workgroup Size for Stencil Patterns." ADAPT 2016.

C. Cummins, P. Petoumenos, M. Steuwer, H. Leather "Towards Collaborative Performance Tuning of Algorithmic Skeletons." HLPGPU 2016.

http://chriscummins.cc

THE UNIVERSITY of EDINBURGH informatics

EPSRC Centre for Doctoral Training in Pervasive Parallelism

EPSRC Engineering and Physical Sciences Research Council

Image credits: "Brain": Department of Molecular & Cell Biology, Berkeley. M82: NASA.

\AM@currentdocname .png

.png