DTU Informatics
Department of Informatics and Mathematical Modeling

Fast High-performance Modeling Tools for Many-core Architectures

Stefan L. Glimberg, Allan P. Engsig-Karup, Hans Henrik B. Sørensen, Nicolai F. Gade-Nielsen, Dennis Noer, Erik Zenner, Morten G. Madsen
gpulab@imm.dtu.dk – http://gpulab.imm.dtu.dk/

Introduction and Background

GPULab - A competence center and laboratory for research and collaboration within academia and partners in the industry. Established in 2008 at the Section for Scientific Computing, DTU Informatics, Technical University of Denmark. In GPULab we focus on the utilization of Graphics Processing Units (GPUs) for high-performance computing applications and software tools in science and engineering. The goals are to contribute to the development of new state-of-the-art mathematical models and algorithms for maximum performance and assimilation of results to academic and industrial partners in our network. Our approaches calls for multi-disciplinary skills and understanding of hardware, software development, profiling tools and tuning techniques, numerical analysis, along with expert knowledge application areas within science and engineering.

Development of a Massively Parallel Wave Analysis Tool

Ongoing work is concerned with the development of a GPU-accelerated nonlinear free-surface model (OceanWave3D) for simulation of unsteady fully nonlinear water waves over uneven depths. The flexible-order finite difference model is based on a unified potential flow formulation, under the assumption of irrotational inviscid flow. We have redesigned the entire algorithm to enable efficient utilization of allocated hardware resources - currently targeting many-core GPUs. Algorithmic efficiency is achieved by solving the bottleneck problem, a large sparse linear system of equations, iteratively by employing a defect correction method preconditioned by a robust multigrid method. This strategy results in more than an order magnitude in both problem size and practical speedup (relative to optimized single-threaded CPU code).

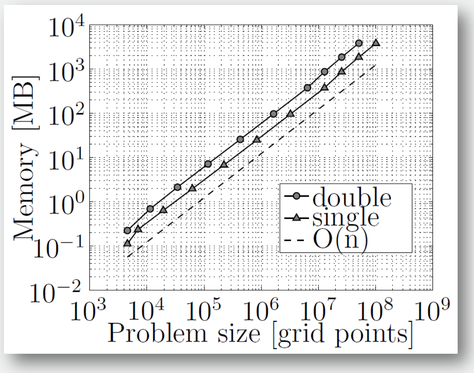


Fig. 1: We achieve linear scaling of the memory footprint for an increasing number of total grid points.

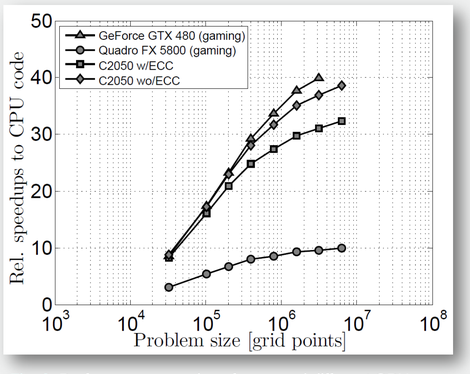


Fig. 2: Performance speedups for several different GPU architectures versus the CPU (single thread) version.

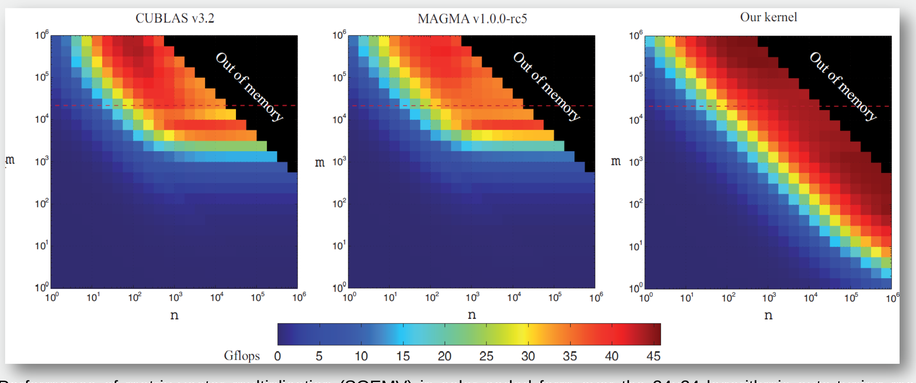


Fig. 3: Performance of matrix-vector multiplication (SGEMV) in color coded form over the 24x24 logarithmic auto-tuning mesh of matrix sizes. Dark blue represents low performance, while dark red represent high performance. The figures compare our auto-tuned kernel to the most commonly used numerical libraries for GPUs, the Nvidia CUBLAS v3.2 and the MAGMA v1.0.0-rc5.

Auto-tuning of Dense Linear Algebra on GPUs

We have implemented an auto-tuning framework that can automate the performance tuning process by running a large set of empirical evaluations to configure applications and libraries on the targeted GPU platform. Preliminary work is focused on dense vector and matrix-vector operations, which form the backbone of level 1 and level 2 routines in the Basic Linear Algebra Subroutines (BLAS) library and are therefore of great importance in many scientific applications. As an example, we develop a single-precision CUDA kernel for the matrix-vector multiplication (SGEMV). The target hardware is the most recent Nvidia Tesla 20-series (Fermi architecture). Our tuned kernels display significantly better performance than the current CUBLAS v.3.2 library.

Fast Cryptanalysis Tool

In this project the focus has been on developing an efficient high-performance tool for crypto-analysis, utilizing affordable many-core consumer Graphics Processing Units (GPUs). The crypto-analysis is based on a bit-sliced DES brute force algorithm. We are developing an efficient implementation of the DES algorithm, which relies mostly on bitwise operations and takes advantage of the high on-chip bandwidth of GPUs. The current implementation is based on CUDA and a GTX 275 gaming card. A break down of the step-wise improvement in the model demonstrates ~10 times speed up in the initial naïve implementation, and after a range of incremental optimizations the implementation achieved a speed up of ~20 times. With the GTX 275 we have found that it is possible to test up to 680 million password keys per second, which is a significant improvement.

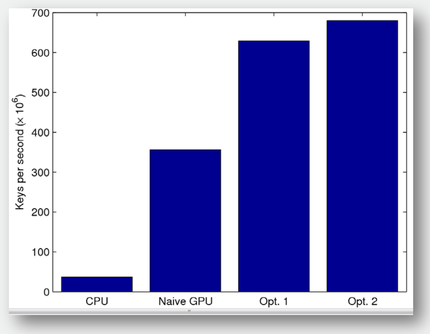


Fig. 4: Efficiency comparison of password key tests per second for CPU, naïve and optimized GPU kernels.

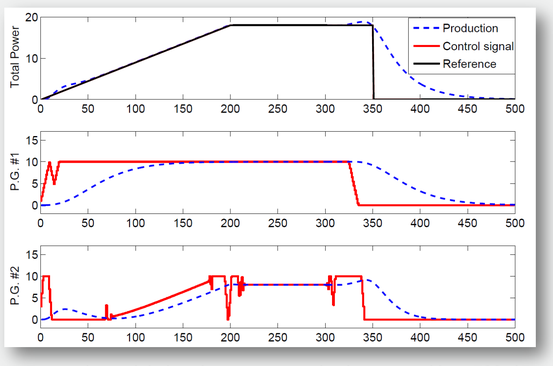


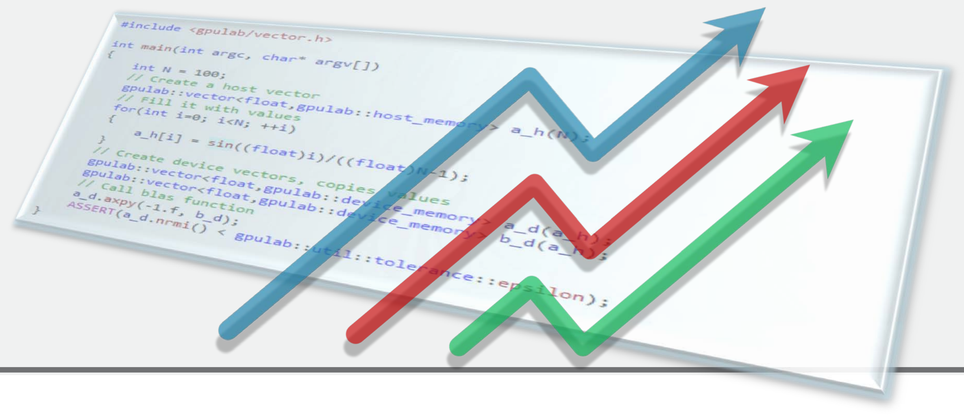
Fig. 5: Power plant control to minimize the energy production cost with a 500 time-step prediction horizon. P.G. #1 is a slow, but cheap power plant while P.G. #2 is a fast, but expensive power plant.

Accelerating Economic Model Predictive Control using GPUs

As stochastic energy production such as wind becomes more common, it is necessary to either store the energy for later consumption or control the energy consumption to coincide with the energy production. One method to address this problem is the Smart Grid, where Model Predictive Control can be used to optimize energy consumption to match with the predicted stochastic energy production and minimize the cost of energy production from conventional power plants. This can be formulated as a convex optimization problem and solved using primal-dual interior-points methods. The main computational tasks in such a method are matrix-matrix multiplications and Cholesky factorization, both of which are very suitable for GPU acceleration. Initial results of a test case controlling two power plants to match energy consumption show an speed-up of up to ~25 using a Nvidia Tesla C2050 compared to a sequential CPU version running on a Intel i7-920.

GPULab Library – a High-performance GPU-based Library for the Development of Scientific Applications

We have an ongoing development of a GPU-based generic C++ library for scientific computing. The two main goals are to create a common playground for the developers at the section and interested network contacts, and to keep an up-to-date platform containing the latest results from the developers. We now have several components for solving large scale partial differential equations. However, this should not be a limitation and we soon expect to have show-cases with dynamic optimization and model control problems as well. In the future we seek to expand the library into a fully distributed tool, in order to achieve maximum performance on cluster-based hardware systems.



GPU-Lab
DTU Informatics

Per Christian Hansen
Jan Hesthaven
Bernd Dammann
John Bagterp Jørgensen

Allan Engsig-Karup
Jeppe Frisvad
Boyan Lazarov
Hans Henrik Brandenborg Sørensen

Stefan Lemvig Glimberg
Nicolai Fog Gade-Nielsen

http://gpulab.imm.dtu.dk/