Enhancing the Efficiency of Massively Parallel Programs in Computational Science and Engineering Applications

Dr. Mayez Al-Mouhamed

http://faculty.kfupm.edu.sa/coe/mayez

Introduction

Massively parallel computing has obtained prominence through advances in implementing massive multithreading and recent improvements in its programming. Recent development in Graphic Processing Units (GPUs) has opened a new challenge in harnessing their computing power as a new general purpose computing paradigm. Strong implications are expected on computational science and engineering, especially in the area of discrete numerical simulation.

Modern GPUs use multiple streaming multiprocessors (SMs) with potentially hundreds of cores, fast context switching, and high memory bandwidth to tolerate ever-increasing latencies to main memory by overlapping long-latency loads in stalled threads with useful computation in other threads. The Compute Unified Device Architecture (CUDA) is a simple C-like interface proposed for programming NVIDIA GPUs. However, porting applications to CUDA remains a challenge to average programmers. CUDA places on the programmer the burden of packaging GPU code in separate functions, of explicitly managing data transfer between the host and GPU memories, and of manually optimizing the utilization of the GPU memory.

In this research we propose to develop a software tool for restructuring C-like loops into optimized CUDA kernels. For this we propose to identify the GPU constraints for maximum performance such that the memory usage (global memory and shared memory), number of blocks, and number of threads per block. In addition we propose to (1) identify the condition for maximizing utilization of the GPU resources, (2) establish the relationships between the influencing parameters, and (3) develop a method for finding possible tiling solutions with coalesced memory access that best meets the identified constraints. Based on the above, we will design a restructuring tool for optimizing performance of CUDA programs. In the evaluation, the above software tool will be used to parallelize the 2-D/3-D Fluid Flow simulation based on the Navier-Stokes Equations for fixed boundary conditions. Obtained performance will be compared to others' contribution.

The above restructuring tool will greatly simplify programming for the best performance of GPU which will contribute to the spreading of the use of GPU supercomputing applications, scientific computing, and more generally the applications of information technology. This research will build sufficient know-how and state-of-the-art tools for the efficient programming of GPUs. This research will stimulate a long-term interest in the research and development of programming massively parallel computers and their applications especially in the Oil and Gas industry. Specifically, the research outcomes will serve the graduate research program and the industry in the kingdom of Saudi Arabia.

Our proposal is to develop software tool to ease the process of writing efficient parallel programs and to use the tool to parallelize the 2-D Fluid Flow simulation based on the Navier-Stokes Equations for fixed boundary conditions. We want to build the expertise and the know-how that will lead to efficiently writing parallel code for scientific simulators.

This research is also about establishing a Massively Parallel Computing Laboratory (MPCL) at KFUPM that will serve the graduate research program in Computer Engineering (COE), Information and Computer Science (ICS), and Mechanical Engineering (ME) at KFUPM. This research will increase the awareness of Massively Parallel Computing among faculty, graduate students, and research assistants at KFUPM.

This research falls under the Saudi national plan of strengthening the sector of information technology, its track on High Performance Computing, and directly addresses its sub-tracks (1) Supercomputing architecture & software and (2) Computer simulation. Our proposal is to develop a restructuring tool to ease the process of writing efficient CUDA programs and to use the tool to parallelize the 2-D Fluid Flow simulation based on the Navier-Stokes Equations. We want to build the expertise and the know-how that will lead to efficiently writing parallel code for scientific simulators to serve the graduate research program and the Oil and Gas industry in the kingdom of Saudi Arabia.

Research Summary

Massively Parallel Computing is gaining ground in high-performance computing. CUDA (an extension to C) is most widely used parallel programming framework for general purpose Graphic processing Units (GPUs). However, the task of writing optimized CUDA programs is complex even for experts. We are proposing to develop an automatic restructuring tool to optimize CUDA programs for computational science and engineering applications with following features:

- Identifying the condition for maximizing utilization of the GPU resources and establishing the relationships between the influencing parameters.

- Developing algorithms that explore possible tiling solutions with coalesced memory access and resource optimizations that best meet the identified restructuring specifications. For this we will tailor the GPU constraints to achieve maximum performance such as the memory usage (global memory and shared memory), number of blocks, and number of threads per block. A restructuring tool (R-CUDA) will be developed to enable optimizing the performance of CUDA programs based on the restructuring specifications.

- Building a 2-D Fluid Flow simulator based on the Navier-Stokes Equations for fixed boundary conditions. The simulator code will be optimized using the above restructuring tool to expose maximum data parallelism in dense and sparse linear algebra solvers.

- Extensive testing of the tool using benchmarks from the LAPACK – BLAS library such as DGEMM, SGEMM, CAXPY and check for correctness. Also the use of profiling tools such as CUDA Visual Profiler, Parallel Nsight, TotalView to verify the restructuring specifications. The simulator will be tested and validated using typical test cases.

The major outcomes of this research are: 1) an automatic restructuring tool for optimizing the performance of CUDA programs focusing on dense and sparse linear algebra solvers, (2) an optimized 2-D Fluid Flow simulator based on the Navier-Stokes Equations for fixed boundary conditions, and (3) a research lab in Massively Parallel Computing and a graduate course in Computational Science and Engineering.

Our proposal is to develop a restructuring tool to ease the process of writing efficient CUDA programs and to use the tool to parallelize the 2-D Fluid Flow simulation based on the Navier-Stokes Equations. We want to build the expertise and the know-how that will lead to efficiently writing parallel code for scientific simulators to serve the graduate research program and the Oil and Gas industry in the kingdom of Saudi Arabia.

References

[1] Seyong Lee, Seung-Jai Min, and Rudolf Eigenmann, OpenMP to GPGPU: A Compiler Framework for Automatic Translation and Optimization, Proc. 14th ACM SIGPLAN Symp. on Prin. and Prac. of Parallel Programming, 2009.

[2] Tianyi David Han and Tarek S. Abdelrahman, “hiCuda: A high-level Directive-based Language for GPU Programming”, GPGPU’09, March 8, 2009.

[3] G. Rudy, "CUDA-CHiLL: A Programming Language Interface for GPGPU Optimizations and Code Generation," MS Thesis, University of Utah. August, 2010.

[4] L. Chen, "Exploring Novel Many-Core Architectures for Scientific Computing," PhD Thesis, University of Delaware. 2010.

[5] Demmel, J. et. al., Self-Adapting Linear Algebra Algorithms and Software, Proc. of the IEEE, Vol. 93, No 2, pp.293-312, 2005.

[6] Tomov, S.; Nath, R.; Ltaief, H.; Dongarra, J., Dense linear algebra solvers for multicore with GPU accelerators,IEEE Inter. Symp. on Parallel & Distrib. Processing, pp. 1-8, 2010.

[7] David B. Kirk and Wen-mei W. Hwu, “Programming Massively Parallel Processors: A Hands-on Approach”, Published by Elsevier Inc. ISBN: 978-0-12-381472-2, 2011.

[8] Igor V. Minin and Oleg V. Minin, Computational Fluid Dynamics Technologies And Applications. Croatia: InTech, 2011.

[9] Johannes Habich, “Performance Evaluation of Numeric Compute Kernels on nVIDIA GPUs” M.S. thesis in informatics,Friedrich-Alexander-University, Erlangen, Germany. 2008.

[10] Mayez A. Al-Mouhamed and Ayaz Khan, Exploration of Automatic Optimization for CUDA Programming, 2nd IEEE International Conference on Parallel, Distributed and Grid Computing, Jaypee University of Information Technology (IEEE-PDGC), Himachal Pradesh, India, 6 December 2012. Selected as the "Second Best IEEE-PDGC-2012 Conference Paper" out of 605 paper submissions.

[11] A. Baqais, M. Assayony, A. Khan, and M. Al-Mouhamed, Bank Conflict-Free Access for CUDA-Based Matrix Transpose Algorithm on GPUs, Accepted in the International Conference on Computer Applications Technology (ICCAT'2013), 22 January, 2013.