Quantifying the Impact of GPU Specific Optimizations: An Experimental Study

A. Saeed*, E. Elwany, E. Tawadros, K. Abdelsalam, P. Yousry, S. Hafez
Departement of Computer and Systems Engineering
Faculty of Engineering, Alexandria University, Egypt

Abstract

When moving from CPU computing to the GPUs domain, different possibilities for optimization appear introducing challenges that include explicit caching, memory addressing and arithmetic accuracy issues along with algorithms' parallelization which is the main motivation. In this work we try to identify some of these problems and quantify the effects they have on the performance.

We chose the problem introduced by John Michalakes et al. [1] to be the basis of our experiments. The problem is concerned with accelerating one of the Weather Research and Forecasting model's (WRF) modules called WRF-Single Moment 5-Class module (WSM5). We translated the most recent version of the WSM5 module [3], originally written in FORTRAN, to CUDA. The translation achieved a 14.6x speedup by only the parallelizing the module's main algorithm.

A series of optimizations were then introduced; starting with arithmetic accuracy optimizations, by using the IEEE compliant arithmetic operations provided by the CUDA Runtime Library. The difference between the results of the FORTRAN code and the CUDA code (i.e. errors) were minimized. Using IEEE compliant arithmetic operations degraded performance, so other optimizations were applied.

Moving extensively used variables to registers and applying loops fusion enhanced the performance, allowing it to be better than before introducing accurate arithmetic operations. Using cache memories was then investigated.

Texture memory is a cached memory space. It uses write through cache coherency protocol [2]. Using texture memory requires explicitly identifying memory regions to be cached and binding those regions to the texture cache through specific directives. We used texture memory to cache read-only variables used throughout the module. This allowed for a 7% speedup allowing for an overall speedup of 15.7x.

To the best of our knowledge, our work is the first to quantify the incremental effects of different GPU specific optimization techniques. This work could be of great help to programmers, as it can give them insight of the expected outcomes of the optimizations to help them weigh the performance gain against the development cost.

1 Eligibility for ACM SRC

All authors of this work are undergraduate students affiliated with Alexandria University. One of the authors is an ACM Student Member.

The author's academic advisors are Prof. Mohamed Abougabal, Prof. Layla Abouhadid and Dr. Ahmed Elmahdy.

References

[1] J. Michalakes and M. Vachharajani, "GPU acceleration of numerical weather prediction", *Parallel Processing Letters*, vol. 18, no. 4, p.531–548, 2008.

[2] N. K. Govindaraju and D. Manocha, "Cache-efficient numerical algorithms using graphics hardware," Parallel Computing, vol. 33, no. 10-11, p. 663–684, 2007.

[3] "GPU acceleration of WSM5 microphysics." [Online]. Available : http://www.mmm.ucar.edu/wrf/WG2/GPU/WSM5.htm

[*] Contact Author ahmed.saeed@acm.org