GPU Group Accomplishments and Status

Ali Akoglu
David Lowenthal
Greg Striemer
Peter Bailey

July 23, 2010

This document summarizes the accomplishments of the GPU group over the period January 7, 2010 through July 15, 2010.

First, we have implemented many important GLM routines in CUDA. Most recently we have implemented forward regression with multiple SNPs. The routines within forward regression have been highly tailored to the GPU architecture. New versions of matrix transpose, multiplication, and inversion have been engineered to deal specifically with the problem of regression in the context of SNPs. By fully customizing each of these routines, we have been able to significantly cut the amount of required computations of the forward regression process, while also creating a massively parallel solver. Other routines which have been customized for the CUDA implementation include the calculation of error and the p-value.

The performance, which was evaluated based on 100K SNPs each of length 191, is excellent—execution time is just 62 milliseconds. The Matlab version of forward regression takes roughly 11 seconds, according to the demo provided by Liya Wang. This means that the GPU speedup is over 177 (with just a single GPU). Furthermore, unlike most GPU-reported results, this time includes the data transfer to the GPU and back to the host machine. (However, it does not include reading data from the input file on the host machine.)

We are currently working on implementing the pairwise version of the regression, and we are talking to Peter Bradbury to acquire a large-scale, real data set. In anticipation of this, we are applying for time on the TACC machine. Forward regression and pairwise regression will be further modified so that the problem can be spread across multiple GPUs to utilize all available parallelism in the TACC system, while also ensuring the performance capabilities of each GPU are fully exploited.

We believe that there are other important routines that can be translated to the GPU. We look forward to studying these routines and subsequently finding a GPU version that obtains 100-fold speedup or more.

Another main accomplishment is a research tool we have been working on to help programmers (and system software) determine where bottlenecks exist in CUDA code. The eventual goal is to assist in the creation of fully-optimized, massively parallel GPU software. The group has submitted a paper on a novel architectural model of GPUs [1]. Specifically, we can analyze a GPU binary and predict performance within 11%. This is much better than previously reported results. This work—once fully mature—will eventually be incorporated into the GLM project. Ideally, this tool will assist us in future GPU routines for the GLM project.

References

[1] G. Striemer, A. Akoglu, D. Lowenthal, P. Bailey, and J. Hartman. A performance modeling tool for graphics processing units. In *Conference on Partitioned Global Address Space Languages* (submitted), July 2010.