Characterizing performance loss from mapping general purpose applications onto GPU architectures

Archit Gupta
Department of Electrical Engineering
and Computer Sciences
University of California, Berkeley
Email: architgupta@berkeley.edu

Sohum Datta
Department of Electrical Engineering
and Computer Sciences
University of California, Berkeley
Email: sohumdatta@berkeley.edu

***Abstract*—The abstract goes here.**

I. Introduction

Graphics Processing Units (GPUs) are increasingly used for scientific and other compute-intensive tasks. Owing to the availability of large number of parallel resources (shader pipelines), they can achieve orders of magnitude higher performance on certain data-parallel applications than a conventional CPU.

However, since the shader cores were not designed for general purpose computing, there is performance loss due to a significant amount of control flow. Additionally, unlike vector processors, most GPUs do not have tightly coupled scalar core(s). Hence, to map general problems on a GPU, the scalar portion of the program can be executed on: (1) the CPU, and the parallel part on the GPU (thereby limiting performance due to inter-system data movement); or (2) the GPU as part of the execution kernel thereby under-utilizing the parallel hardware avaliable. The latter approach causes the inclusion of significant amount of control flow in existing kernels. Our objective is to investigate their effects on overall performance. December 12, 2015

II. Overview of Graphic Processing Units

III. Problem Description

IV. Related Work

V. Our Approach

VI. Results

VII. Conclusion

Acknowledgment

The authors would like to thank...

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.