Hierarchically Characterizing CUDA Program Behavior

Zhibin Yu, Hai Jin
Service Computing Technologies and System Lab/Cluster and Grid Computing Lab, Huazhong University of Science and Technology, Wuhan, China
{yuzhibin,hjin}@hust.edu.cn

Nilanjan Goswami, Tao Li
Intelligent Design of Efficient Architecture Lab, University of Florida, Ganiesville Florida, USA
nil@ufl.edu, taoli@ece.ufl.edu

Lizy K. John
Dept of Electrical and Computer Engineering, University of Texas at Austin Austin, TX 78712 USA
ljohn@ece.utexas.edu

Abstract

CUDA has become a very popular programming paradigm in parallel computing area. However, very little work has been done for characterizing CUDA kernels. In this work, we measure the thread level performance, collect the basic block level characteristics, and glean the instruction level properties for about 35 programs from CUDA SDK, Parboil, and Rodinia benchmark suites. In addition, we define basic block vectors, synchronization vectors and thread similarity matrix to capture the characteristics of CUDA programs efficiently. We find that CUDA programs have some unique characteristics at each level compared to sequential programs.

1. Introduction

Compute Unified Device Architecture (CUDA) programming mode is very different from sequential programming modes. To characterize CUDA program behavior and understand why and where they can achieve significant speedup comparing to sequential programs, it is important to revisit the basic block level and instruction level properties besides those at the thread level. In this paper, we propose to characterize CUDA program behaviors hierarchically by quantitatively gleaning properties from thread, basic block, and instruction levels.

In addition, previous researchers have demonstrated that basic blocks vectors (BBVs) are one of the most accurate techniques for creating code signatures [1] for sequential programs. In this paper, we firstly employ basic block and basic block vectors to analyze the code signature of CUDA threads. We observed that basic block characteristics of CUDA kernels are very different from those of sequential programs. Based on the basic block vectors, we construct the similarity matrix of threads. We show that the similarity matrix can be a very powerful tool for performance tuning.

2. Preliminary results

Compared to the traditional sequential benchmarks such as SPEC CPU2000, MiBench, and MediaBench, the number of static basic blocks of CUDA kernels is 11~25 times less than that of the aforementioned sequential programs.

At thread level, we find that the performance of CUDA thread is extremely low compared to the thread running on CPU processors. In addition, we observe that there is a significant imbalance among the threads.

At instruction level, the percentage of distances less than 8 is almost 60%. However, there are still a large amount of large instruction dependency distances. Precisely, more than 20% of instruction dependency distance is larger than 8. This indicates that we still have relatively high opportunities to harness the instruction level parallelism of arithmetic instructions to improve the performance of GPGPU processors.

Our similarity matrix is constructed from the basic block vectors of the CUDA threads. We generate the similarity matrix figure by using darker color to represent more similarity among threads and vice versa. From the figures, it is very easy to identify where we can do more optimizations in CUDA kernels.

3. References

[1] A. S. Dhodapkar and J. E. Smith, “Comparing Program Phase Detection Techniques”, *in Proceedings of the 36th International Symposium on Microarchitecture (MICRO-36)*, IEEE Computer Society, San Diego, CA, USA, December, 2003, pp. 217-227.