ECS 158 Final Report:
combn() in Snow, OpenMP, and CUDA

Stefan Peterson
Bijan Agahi
Arjun Bharadwaj

Abstract

Parallel architectures are becoming increasingly prevalent in the modern world and as such have spawned multiple frameworks to assist with parallel programming, but these frameworks vary greatly in their implementation and performance. This paper takes a close look at three frameworks: R-Snow, OpenMP and CUDA. We took on the task of re-writing the combn() function found in R using each of these frameworks. We measure run times of these tests at different sizes and with varying inputs and levels of difficulty. In the end, we point out the strengths and weaknesses of each language as well as note on features which we, as programmers, find particularly useful or troublesome.

Contents

1 Introduction

With the relatively recent introduction of widely available multi-processor CPUs in household computers, parallel programming has become a hot topic in the world of computer science. In particular, the graphics and games industry has driven the multithreaded programming style to the forefront of everyone's attention. Through the process of splitting up tasks and distributing those tasks to individual cores on the CPU or GPU, parallel processing in most cases can provide vast improvements in both performance and efficiency. OpenMP, R-Snow, and CUDA are three such parallel languages, and this paper aims to compare and contrast the strengths, weaknesses, and performance of each language.

2 Data / Timing Experiments

stuff here

2.1 Setup for Benchmarks

All tests for the purposes of this paper were run on Tetraat University of California, Davis. This computer consists of a 64-bit 8-core processor with 16 GB or RAM and an Intel Core i7-2600K CPU with a clock speed of 3.4 GHz and a cache size of 8192 KB. The machine was running release 19 of the linux distribution Fedora.

2.2 combn() in R-Snow

2.3 Graphs and Tables

graphs and tables go here

2.4 combn() in OpenMP

stuff here

2.5 Graphs / Table

graphs and tables here

2.6 combn() in CUDA

stuff here

2.7 Graphs / Table

graphs and tables here

3 Analysis

3.1 Finding triples in adjacency matrix

4 Research Papers

4.1 Authors — Position

4.1.1 Description

4.1.2 Analysis

5 Conclusion

stuff here

6 Bibliography

A Appendix: Code

going to add code snippets here

There are also a variety of files specific for testing purposes included in the submitted .tar file.

B Appendix: Member Contributions

Stefan Peterson

- LaTeX Formatting, editing
- OpenMP implementation
- Abstract
- Introduction
- Research paper summarization/analysis
- Conclusion
- Data analysis writeup
- Relevant research

Bijan Agahi

- R-Snow implementation
- Experiment data collection
- Test scripting
- Graphing in R, tabling
- Relevant research

Arjun Bharadwaj

- Thrust implementation
- Relevant research