Stable Matching with CUDA

June 5, 2018

Abstract

I plan to explore the best way to do stable matching with CUDA, with a specific focus on implementing PII in CUDA.

1 Objectives and Methods

My primary objective is to compare different ways of doing stable matching in CUDA. I plan to implement several stable matching algorithm in CUDA in order to compare them such as the ones mentioned in [5]. However, I will primarily focus on implementing PII as described in [4] and PII-SC as described in [7]. [3] claims to improve on PII-SC so I'd like to find more information on those improvements and consider implementing them.

While I'm implementing PII-SC I plan to look for ways to improve it. I'd love to be able increase the probability of convergence and deal with some of the problematic cases described in [7]. I think being able to improve PII-SC would be a very interesting result, but I'm not confident in my ability to do it.

2 Methods

I plan to implement CUDA versions of the stable matching algorithms described in [2] and [6] aided by the work in [5]. If I discover any other stable marriage algorithms that appear to lend themselves to a CUDA implementation I will implement those as well.

When I say I plan to implement PII and PII-SC on CUDA, I mean I plan to implement the mathematical method underlying PII and PII-SC as described in section 2 of [4]. I plan on using a implementation that departs dramatical from the fully connected PRAM computation model used in section 3 of [4]. [1] clearly demonstrates that implementing PII on CUDA as though CUDA was a fully connected PRAM produces slow results in practice.

I plan to experimental compare runtimes for the CUDA stable matching solvers I end up with. I'm not sure what the standard methods of measuring and comparing runtimes of CUDA programs or stable matching algorithms are so I will need to do further research that subject.

3 Timeline

Week 2 (staring June 5):
In this week I plan to make multiple CUDA versions of the algorithms described in [2] and [6] with the help of [5]. I want to implement these simple CUDA algorithms first before I dive into something complicated like PII. This week will also give me an opportunity to become more familiar with the CUDA architecture and optimization on CUDA. This week I will also begin to develop methods for testing stable marriage algorithms on CUDA.
Week 3 (starting June 12):
In this week I plan to implement PII on CUDA.
Week 4 (starting June 19):
In this week I plan to tune my PII CUDA implementation, and implement PII-SC on CUDA. If the improvements [3] claims to have generated turn out to be helpful I will implement those as well.
Week 5 (starting June 26):
Either explore ways to improve PII-SC further or find other stable matching algorithms to implement in CUDA depending on which seems more promising.
Week 6 (starting July 3):
Perform experiments, gather data, and start writing.
Week 7 (starting July 10):
Explore whatever aspect of the project seems most promising when I get to this point.
Week 8 (starting July 17):
Explore whatever aspect of the project seems most promising when I get to this point.
Week 9 (starting July 24):
Finish writing.

References

[1] Andrew Barkley and Justin Martin. *Parallel Iterative Improvement Stable Matching Algorithms.* http://faculty.salisbury.edu/~ealu/reu/Projects_File/2017/Lu.pdf.

[2] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[3] Blake Hurlbut, Ryan Yost, and Lu Enyue. *Parallel Iterative Improvement Stable Matching Algorithms.* http://faculty.salisbury.edu/~ealu/reu/Projects_File/2017/Lu.pdf.

[4] Enyue Lu and S. Q. Zheng. A parallel iterative improvement stable matching algorithm. In Timothy Mark Pinkston and Viktor K. Prasanna, editors, *High Performance Computing - HiPC 2003*, pages 55–65, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[5] Fredrik Manne, Md. Naim, Håkon Lerring, and Mahantesh Halappanavar. *On Stable Marriages and Greedy Matchings*, pages 92–101.

[6] D. G. McVitie and L. B. Wilson. The stable marriage problem. *Commun. ACM*, 14(7):486–490, July 1971.

[7] Colin White and Enyue Lu. *An An Improved Parallel Iterative Algorithm for Stable Matching.* personal communication from Enyue Lu. (The longer form of Colin's paper retrieved from MyClasses).