Fast Poisson Image Blending using Parallelized Jacobi Method

Amatur Rahman
Pennsylvania State University

Introduction

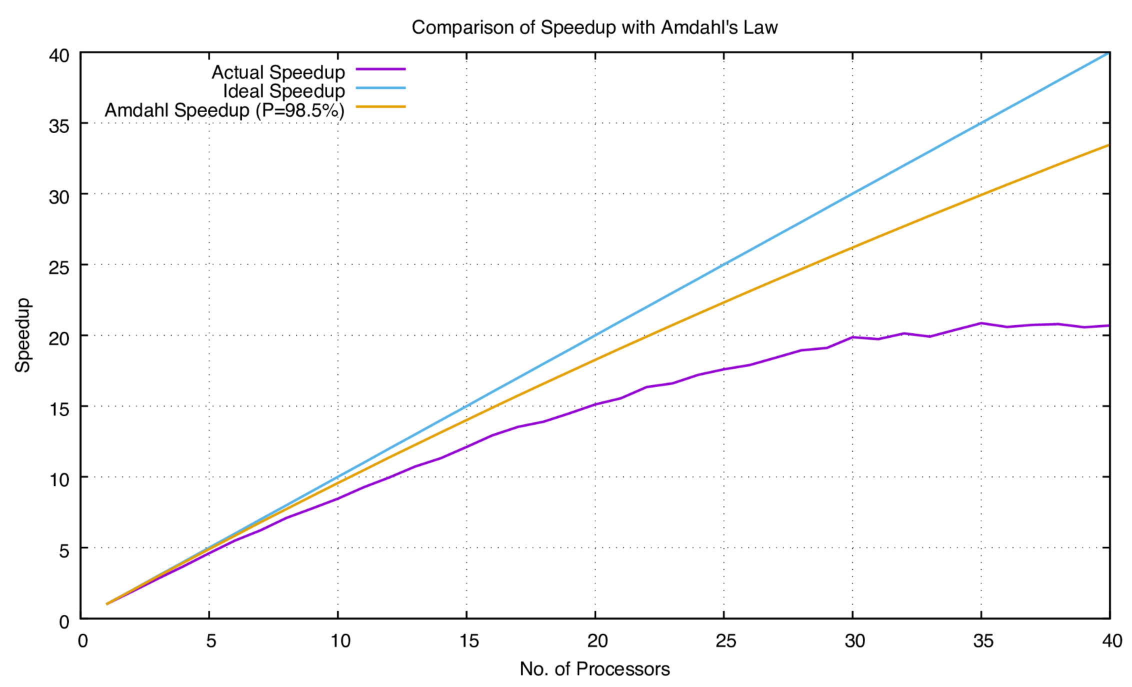
Add your information, graphs and images to this section.

Problem Description

Add your information, graphs and images to this section.

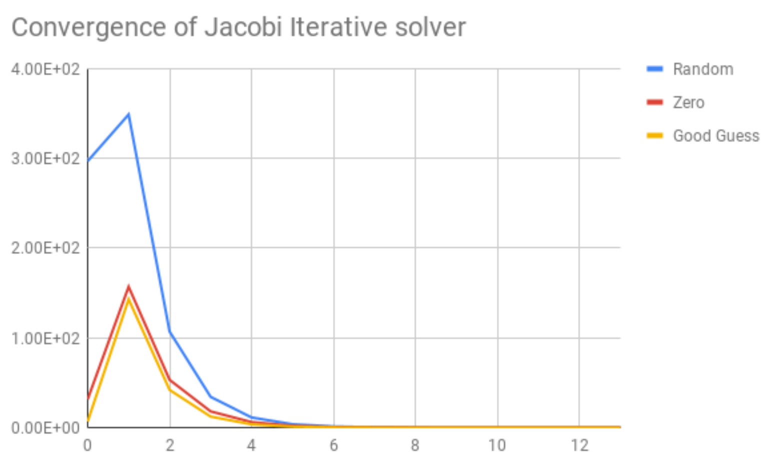
Effect of Parallelizing the Code

Add your information, graphs and images to this section.



Direct vs. Indirect Solvers

- Our 'A' matrix is sparse, so indirect solver Jacobi does better.
- Sparse matrices do not generally have sparse LU decomposition, so it gets harder to fit the matrices in memory as the problem size goes larger.
- For Jacobi method, we can just store the A matrix in sparse representation.
- We tweaked the initial conditions to achieve fast convergence of direct solver Jacobi.



Strong Scaling

Add your information, graphs and images to this section.

Results Overview

Add your information, graphs and images to this section.

Coupling

Add your information, graphs and images to this section.

Acknowledgements

ICS-ACI computing resources were used to perform the computations.

All resources were provided by Dr. Adam Lavely and Dr. Chris Blanton.

Code Availability

The code is publicly available in github:
https://github.com/amatur/cse597_parallel_solver

Bibliography

1] Poisson blending. Retrieved from http://eric-yuan.me/poisson-blending/ accessed 2018-09-23.
[2] Poisson image editing. Retrieved from http://www.ctralie.com/Teaching/PoissonImageEditing/, accessed 2018-09-23.
[3] Barker, B. Message passing interface (mpi). In Workshop: High Performance Computing on Stampede (2015), vol. 262.