Parallel Implementation of Linear System Solvers

Weiming Hu*
Geoinformatics and Earth Observation Laboratory
Dept. of Geography and Institute for CyberScience
The Pennsylvania State University
*weiming@psu.edu**

Introduction

Parallelization
Application in weather model output statistics

Serial and Parallel Implementation

Shared memory parallelization: OpenMP
Distributed memory parallelization: OpenMPI

Direct and Iterative Solvers

Direct solver: normal method

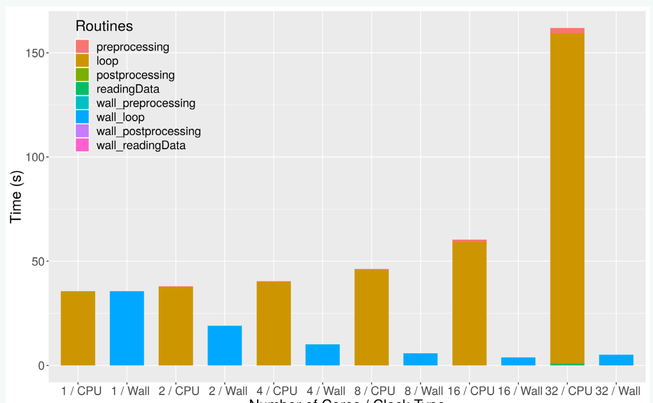
There are three popular ways to solve a linear system: 1)Cramer's Rule; 2) Gaussian Elimination; 3)LU Decomposition. The Cramer's Rule is the most inefficient method out of the three. For a linear system with n equations and n unknowns, the Cramer's Rule requires to solve n + 1 different determinants assuming all the determinants exist. However, calculating determinants should be avoided in all efforts because it is a very computationally expensive task whose computational complexity is estimated to be polynomial[4]. Gaussian Elimination and LU Decomposition are usually integrated to solve multiple linear systems with higher efficiency. Calculating determinants can be avoided in these methods by using forward elimination and backward substitution. LU Decomposition involves one more step, the forward substitution, than Gaussian Elimination, leading to the result that LU Decomposition might take longer time to solve one linear system than Gaussian Elimination. However, Gaussian Elimination modifies the coefficient matrix and the right- hand vector internally within each loop, making it impossible to save the transformation and the factorization information of the coefficient matrix. When presented with problems of multiple linear systems where the coefficient matrix does not change and the right-hand vector changes, LU Decomposition has the advantage to save the factorization and reuse the factor matrices for different right-hand vectors. Complexity-wise, Gaussian Elimination is O(nm3) and LU Decomposition is O(m3 + nm2) where n is the number of linear systems to solve and m is the size of the problem.

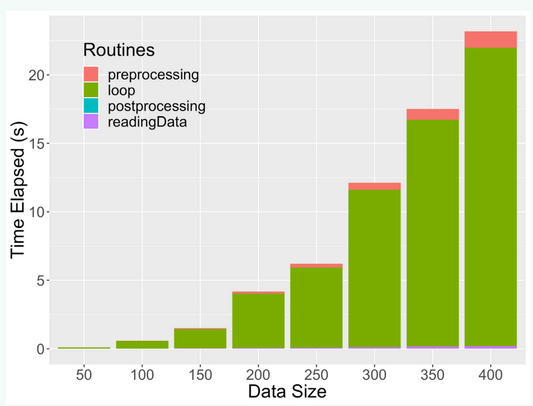
Iterative solver: Jacobi method

Popular iterative solvers come from 1) the Jacobi method and 2) the Gauss-Seidel method. They uses the similar iteration scheme. The scheme can be expressed as $x_{k+1} = M^{-1} \times (b - N \times x_k)$, where x is the solution vector, b is the right-hand-side vector, and M and N matrices are defined so that $A = M + N$. The difference of Jacobi and Gauss-Seidel methods lies in how they define the decomposition matrices M and N. Jacobi method simply define them using the identity matrix. This is a stable approach that the algorithm convergence is guaranteed. However, the convergence rate is very low in practice. The Gauss-Seidel method defines the decomposition using the upper and lower strict triangular parts of the matrix A. This method has a faster convergence.

Profiling

Shared memory parallelization: OpenMP
Distributed memory parallelization: OpenMPI





Access Github Repo

Weather Model Output Statistics

Analog Ensemble explained.
Example results shown.

Acknowledgments

I would like to express my gratitude to the instructors of CSE 597, Dr. Adam Lavely and Dr. Christopher Blanton for the excellent organization and guidance for the course. I would like to also thank my advisor, Prof. Guido Cervone, for mentoring and guiding me.