Parallel Implementation of the Jacobi Method

Weiming Hu*

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

Introduction

Weather Model Output Statistics (MOS) are computationally expensive due to the large amount of the weather model output. Because weather models are initiated with perturbed parameters and run for multiple times, the output contains an ensemble of predictions which are used to estimate the model uncertainty subsequently. Ensemble members contribute differently to the final forecast.

This poster presents a parallel implementation of the Jacobi iterative solver and its application in weather model output statistics to analyze the contribution of each ensemble member. Two parallelization schemes are designed and compared with the industrial de-facto standard libraries OpenMP and OpenMPI. The profiling is carried out on Penn State ICS ACI clusters.

Direct and Iterative Solver

Direct solver is included in the design and implemented for comparison and debugging purposes, though it is intrinsically hard to parallelize. The Normal Equation is chosen as the direct solver because it has a wide range of application in both square and non-square matrices. For a linear system Ax=b, the Normal equation provides the following solution:

$$x = A^t \times \left(A \times A^t\right)^{-1} \times b$$

The Jacobi method is used to design the iterative solver. When the change of the solution is below a certain predefined small value, the iteratio nis terminated. The iterative scheme can be expressed as follows:

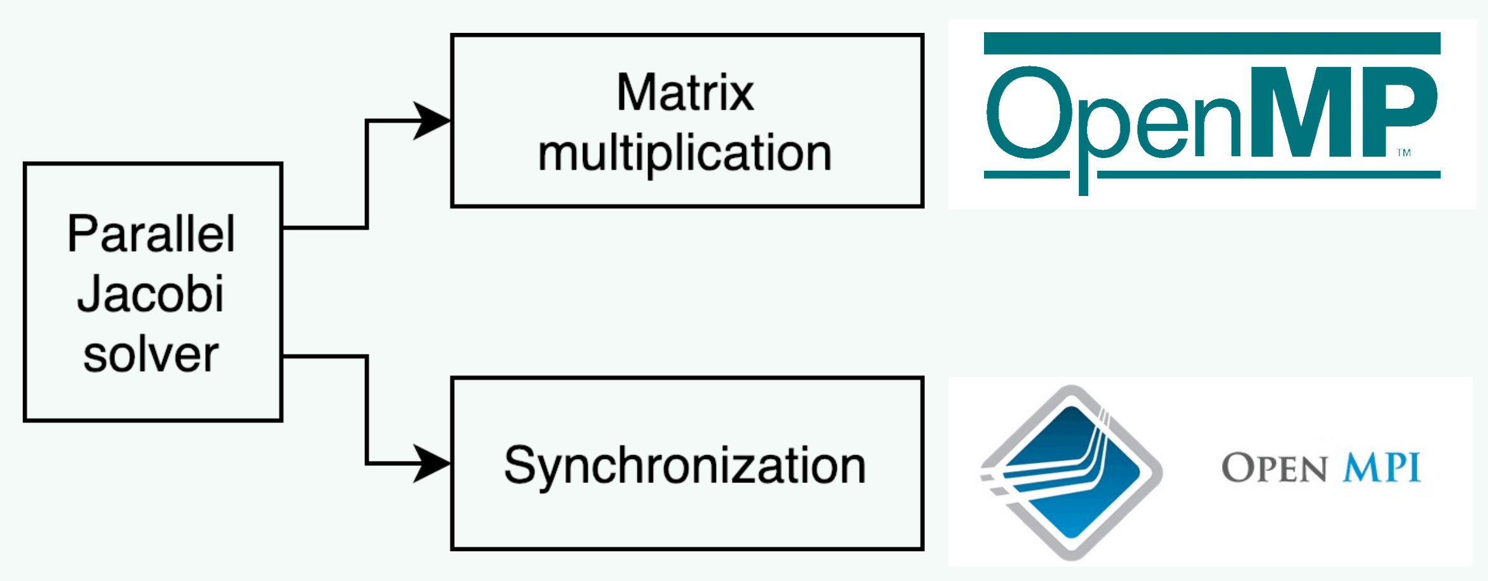
$$x_{k+1} = D^{-1} * (b - A * x_k)$$

Matrix Setup

$$\begin{bmatrix} - & F_1^n & - \\ - & F_2^n & - \\ \vdots & \vdots & \vdots \\ - & F_i^n & - \\ \vdots & \vdots & \vdots \\ - & F_m^n & - \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} O_1 \\ O_2 \\ \vdots \\ O_i \\ \vdots \\ O_m \end{bmatrix}$$

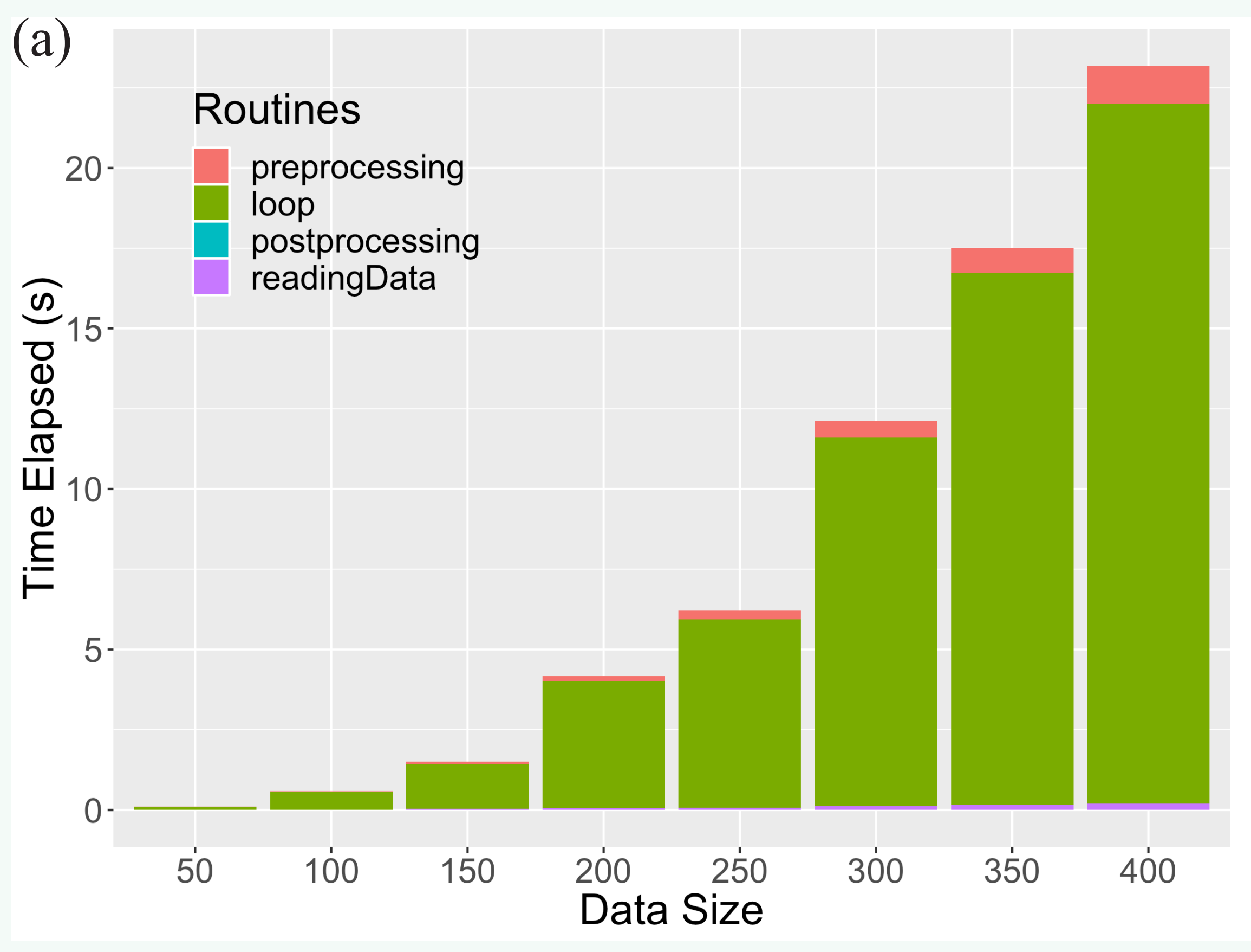
Matrix F is the ensemble forecast with n members and m grids. The 2-D space is broken down to a 1-D vector of grid points. Vector O is the observation for each of the grid points. Vector x measures the correlation between each ensemble member and the correspond observation averaged across the region of interest which makes the solution a spatial metric for the ensemble forecast.

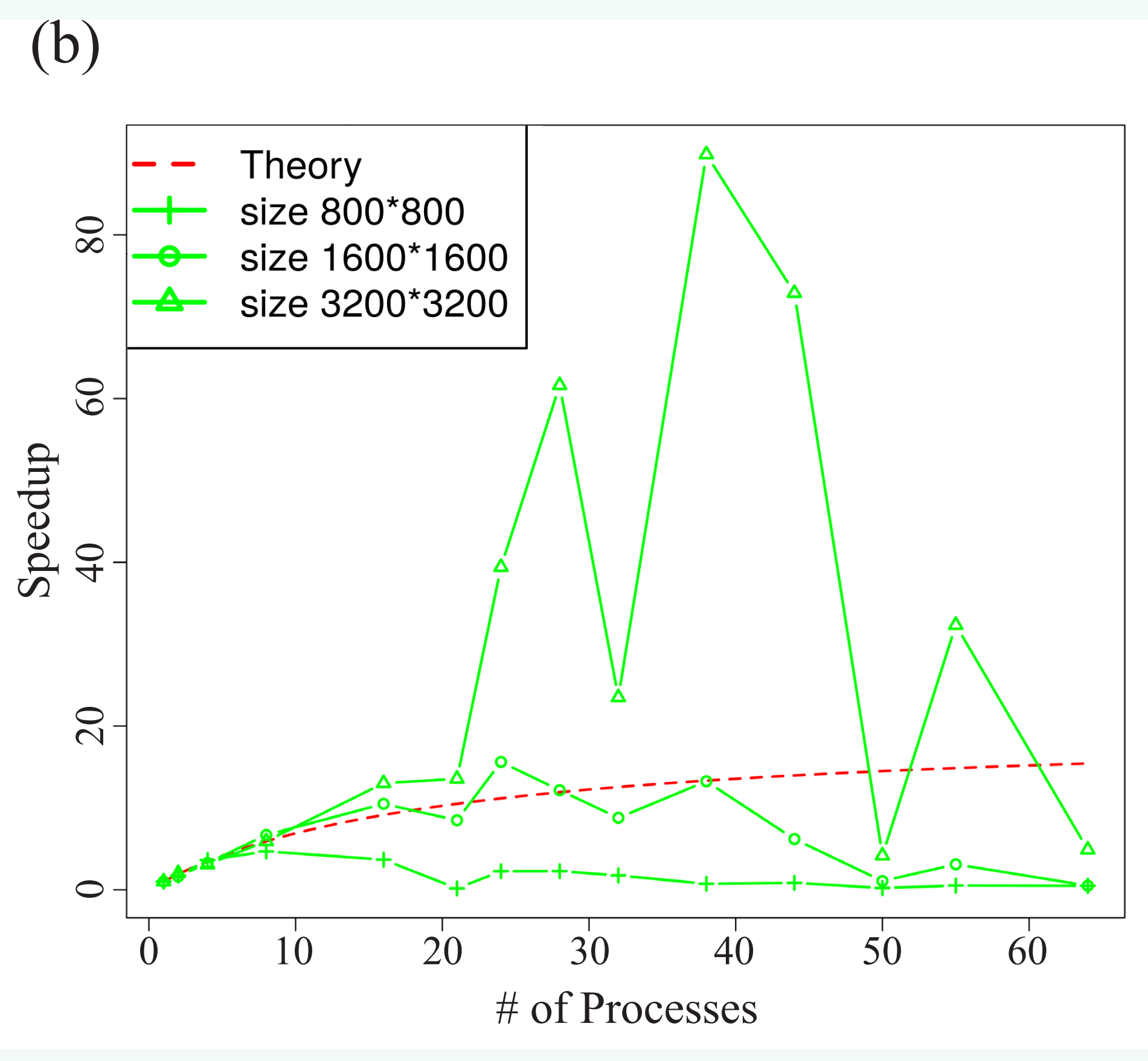
Parallel Implementation

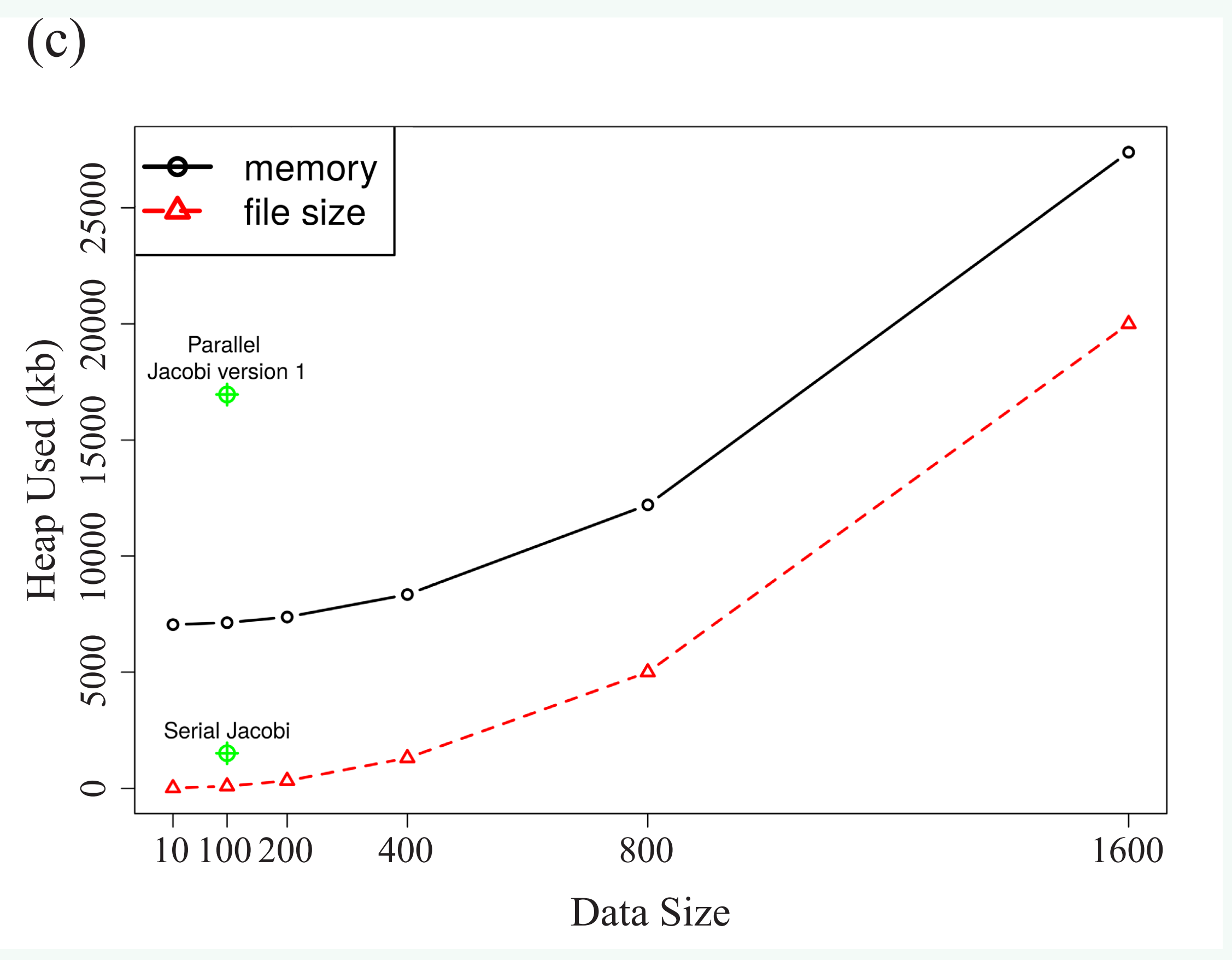


Data are distributed and synchronized using the OpenMPI library because each node only reads its own share of the full data set and, at the end of each iteration, the solution needs to be updated on all nodes. Most of the computation happens during matrix multiplication. This is parallelized using OpenMP because matrix are loaded into shared memory on a single node.

Results and Profiling







Profiling is carried out on ACI ICS basic computing nodes. All experiments are repeated at least 3 times for statistical accuracy and then records are averaged. Test data matrices are synthesized and squre.

Fig. (a) shows the profiling of subroutines in the parallel Jacobi solver as a function of the data size. Most of the time are spent during iterations. The growth of the execution time is proportional to the growth of data sizes.

Fig. (b) shows the speedup of the parallel Jacobi solver with different test data sizes as a function of the number of processors used. A comparison speedup curve with 95% parallelization is calculated based on Amdahl's Law. The speedup becomes more notable when a larger test data set is used. Spikes can be observed for the curve with data size 3200*3200. This is because the distributed data can be effectively padded into caches which leads to super-scaling.

Fig. (c) shows the memory profiling as a function of data sizes. Parallel Jacobi version 1 uses fenced functions to facilitate remote memory access. However, this is proved to be slower and less efficient than the broadcast-and-gather model.

Future Work

1. Scientific application of the Jacobi solver in MOS;
2. Further coupling of the program with other methods, for example, the Gauss-Seidal method.

Access Jacobi solver

Access AnEn package

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.

Hong, Changwan, et al. "Effective padding of multidimensional arrays to avoid cache conflict misses." ACM SIGPLAN Notices. Vol. 51. No. 6. ACM, 2016
Margaris, Athanasios, Stauros Souravlas, and Manos Roumeliotis. "Parallel implementations of the jacobi linear algebraic systems solve." arXiv preprint arXiv:1403.5805 (2014).