Parallelization with Ordered Sweep and Domain Decomposition in 2D and 3D S_N Method

Yos P. Sitompul and Nam Zin Cho*
Korea Advanced Institute of Science and Technology
Corresponding author: nzcho@kaist.ac.kr

1. Introduction

The procedure that requires most of the calculation time in S_N method is the source iteration inversion that is also called sweeping. This calculation can be speeded up by utilizing parallel computers. Many of parallel algorithms that have been developed were coarse grained [1-4], until the ordered sweep was introduced [5]. The ordered sweep algorithm based on diagonal line sweep offers utilization of massive computers and also retains the ability to invert the source iteration in a single sweep. The algorithm itself can be improved with the combination with other parallel algorithms. The performance of the ordered sweep algorithm combined with the domain decomposition is described in this paper.

2. Methods

In this section some of the techniques used to make parallelization of spatial domain are described. It includes the ordered sweep algorithm, the domain decomposition algorithm, and the combination of the ordered sweep and domain decomposition algorithm.

2.1 Ordered Sweep

The ordered sweep algorithm is based on the diagonal line/plane sweep method. In two-dimensional geometry, all cells along the diagonal line can be calculated simultaneously. The sweep progression is shown in Fig.1a. In three-dimensional geometry, all cells along the diagonal plane can be calculated simultaneously.

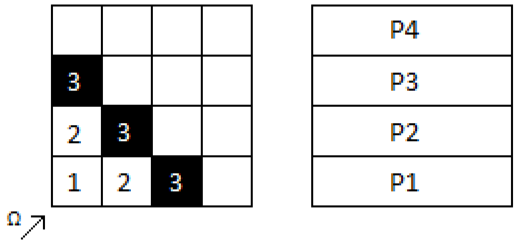

Fig. 1. The ordered sweep (a) sweep progression (b) spatial domain assigned to four processors

The ordered sweep algorithm maps this diagonal line/plane into the processors grid. In two-dimensional geometry, the diagonal line is projected onto *Y* line (can be *X*), shown in Fig.1b, so that the parallelization occurs in this line while the other is performed serially. In three-dimensional geometry, the diagonal plane is projected onto *Y-Z* plane (can be *X-Y* or *X-Z*).

This algorithm has the following characteristics. It represents the intrinsic S_N sweep and therefore no degradation in iteration convergence. However, it produces idle processors during startup and completion, reducing the parallel computational efficiency. To overcome the inefficiency, the algorithm must be able to pipeline the sweep. Method 1 is to pipeline the angles within one octant and also the pair of octant that is calculated serially, i.e., the $\mu < 0$ octant and the $\mu > 0$ octant, while Method 2 is to pipeline the pair of octant only.

2.2 Domain Decomposition

The domain decomposition is decomposing the spatial domain into several subdomains that can be processed concurrently, shown in Fig.2a. The decomposition can be done in one axis or in every axis. For example in two-dimensional geometry, the decomposition can be *X*, *Y*, or *X-Y* decomposition, shown in Fig.2b. For three-dimensional geometry, the decomposition can be *X*, *Y*, *Z*, *X-Y*, *X-Z*, *Y-Z*, or *X-Y-Z* decomposition.

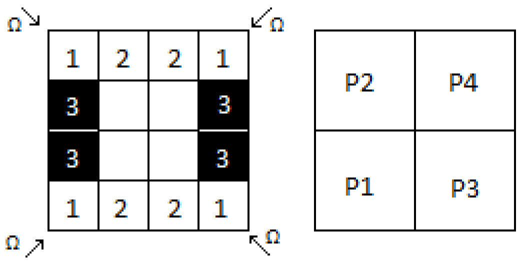

Fig. 2. The domain decomposition (a) sweep progression (b) spatial domain assigned to four processors

At the beginning of a transport sweep, each processor has estimates for the incident interface fluxes for its subdomain. At the end of a transport sweep, new estimates have been calculated and the exiting interface fluxes are shifted to neighboring processors. The transport sweep must be ordered so that each processor is able to use the new outgoing interface boundary fluxes from its neighbors as soon as possible. This is called the alternating direction sweeps.

This algorithm has the following characteristics. It does not produce idle processors and it decomposes the spatial domain in every axis. However, there is degradation in the convergence rate due to the geometric domain decomposition, thus reducing the parallel efficiency. The degradation is not significant provided the subdomains assigned to separate processors do not become optically thin.

2.3 Ordered Sweep and Domain Decomposition

The algorithm decomposes the spatial domain into several subdomains and performs the ordered sweep, based on diagonal line sweep, within each domain. The domain decomposition is applied in subdomain level using the alternating direction sweep, shown in Fig.3.

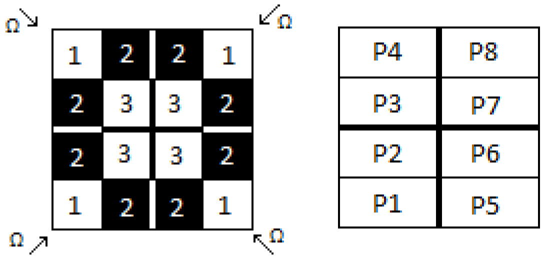

Fig. 3. The ordered sweep – domain decomposition (a) sweep progression (b) spatial domain assigned to eight processors

At the beginning of transport sweep, each processor has estimates for the boundary conditions. The sweep begins from the corner having all boundary conditions required within each subdomain. After one sweep inside the subdomain, the exiting interface fluxes are shifted to neighboring subdomain. Then sweep for other octants until every octant has been swept.

This algorithm has the following characteristics. Since the domain decomposition is applied in larger subdomain, the convergence degradation is small. The ordered sweep is applied inside the subdomain and it decreases the number of idle processors, thus improving the efficiency of the basic ordered sweep. The domain decomposition also allows decomposition in every axis, thus giving the possibility to use more processors compared to the basic ordered sweep.

3. Problem Test and Results

The problem tested for two-dimensional geometry is a one group, isotropic, fixed source problem. All calculations were performed using 320 × 320 mesh, S_{16} product quadratures, and relative error tolerance of 10^{-4}. The serial calculation finished in 1599 iterations and took 2081.43s. The problem tested for three-dimensional geometry is a one group, isotropic, eigenvalue problem. All calculations were performed using 120 × 120 × 120 mesh, S_4 product quadratures, and relative error tolerance of 10^{-7}. The serial calculation finished in 387 iterations and took 2565.46s.

Fig.4. shows the speedup for the two-dimensional geometry. The speedup of the spatial domain decomposition is shown to be higher than the angular domain decomposition (ADD). This is because in our cluster system, each processor communicates through the master node, makes the communication processed in turn and costly. The angular domain decomposition requires each processor to transfer scalar flux information which is proportional to mesh size, while the spatial domain decomposition can handle this with the blocking concept. Fig.5. shows the speedup for the three-dimensional geometry.

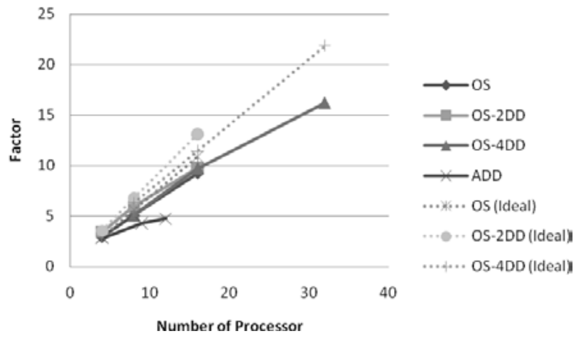

Fig. 4. Speedup for 320 × 320 Mesh (Method 2).

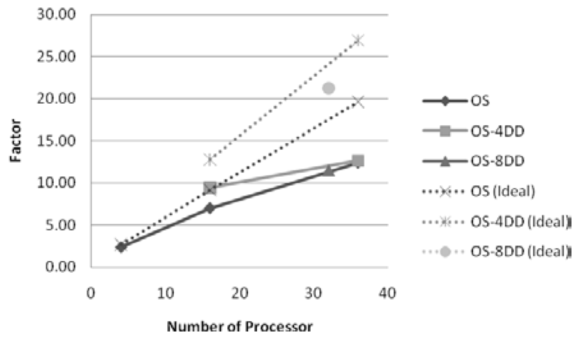

Fig. 5. Speedup for 120 × 120 × 120 Mesh (Method 2).

4. Conclusions

The combination of the ordered sweep and domain decomposition shows the scalability and increases the performance of the basic ordered sweep in speedup and the number of processors that can be utilized. This is achieved by improving the parallel computational efficiency and decomposing the spatial domain in every axis.

REFERENCES

[1] R. Mattis, A. Haghighat, Domain Decomposition of a Two-Dimensional S_N Method, Nuclear Science and Engineering, Vol.111, p.180, 1992.

[2] A. Haghighat, M. A. Hunter, Iterative Schemes for Parallel SN Algorithm in a Shared-Memory Computing Environment, Nuclear Science and Engineering, Vol.121, p. 103, 1995.

[3] A. Haghighat, Angular Parallelization of a Curvilinear S_N Transport Theory Method, Nuclear Science and Engineering, Vol.108, p. 267, 1991.

[4] M. Yavuz, E. W. Larsen, Iterative Methods for Solving x-y Geometry SN Problem on Parallel Architecture Computers, Nuclear Science and Engineering, Vol.112, p. 32, 1992.

[5] R. S. Baker, K. R. Koch, An S_N Algorithm for Massively Parallel CM-200 Computer, Nuclear Science and Engineering, Vol.128, p312, 1998.

[6] E. E. Lewis, W. F. Miller, Computational Methods of Neutron Transport, John Wiley & Sons, New York, pp.156-174, 1984.