Angular Direction Parallelization on STRAUM code using OpenMP

Seungil Jeong , Ser Gi Hong *
Department of Nuclear Engineering, Hanyang University, 222, Wangsimni-ro, Seongdong-gu, Seoul, Republic of Korea
**Corresponding author: hongsergi@hanyang.ac.kr*

1. Introduction

In many areas of nuclear engineering, accurate solutions to the radiation transport equation are required for complex geometry problems. Monte Carlo method can give accurate solutions for this kind of problems via stochastic simulations of the particles but it still requires long computing time to obtain flux solutions in fine energy-spatial meshes even with modern parallel computation. The discrete ordinate(S_N) method, which solves the transport equation by discretizing spatial and angular variables can treat the complicated geometries using unstructured meshes [1,2] generated based on CAD modeling.

The STRAUM(S_N Transport for Radiation Analysis with Unstructured Meshes) code is a deterministic code using the S_N method and unstructured tetrahedral meshes [2,3,4]. The purpose of the STRAUM code is to provide the precise forward and adjoint flux solution results of the transport equation for complicated geometrical problems [3]. Recently, STRAUM has employed the Krylov subspace methods (e.g., Bi-CGSTAB and GMRES) with DSA and TSA preconditioners, and particularly the conventional Gauss-Seidel iteration for upscattering groups was replaced with the Krylov subspace methods [4]. However, the transport sweeping calculations take most of the computing time and so the reduction of the computing time in transport sweeping is essential to further reduce the computing time. In this work, the parallel computations on transport sweeping along the angular directions for STRAUM are studied using OpenMP library based on shared memory [5]. In particular, the performances of parallelization are tested for a quarter reactor pressure vessel problem [4].

2. Methods and Results

2.1 Boltzman Trasnport Equation

In this section, the methodology used in STRAUM is briefly reviewed. The steady state multi-group Boltzmann Transport Equation is given by

$$\begin{aligned}&\widehat{\Omega}_n \cdot \nabla\psi_n^{\mathrm{g}}(\vec{r}) + \sigma_t^{\mathrm{g}}(\vec{r})\psi_n^{\mathrm{g}}(\vec{r}) \\ &= \sum_{\mathrm{g}'=1}^{G}\sum_{\ell=0}^{L}\sum_{m=-\ell}^{\ell}(2\ell+1)\sigma_{s,\ell}^{\mathrm{g}'\to\mathrm{g}}R_{\ell m}(\widehat{\Omega})\phi_{\ell m}(\vec{r}) \\ &\quad + q_{ex,n}^{\mathrm{g}}(\vec{r}),\end{aligned} \tag{1}$$

where ψ_n^{g} and σ_t^{g} represent the angular flux for $\hat{\Omega}_n$ and total macroscopic cross section for group g at $\vec{r}$, respectively, and $\sigma_{s,\ell}^{\mathrm{g}'\to\mathrm{g}}$ represents the ℓ'th moment of the scattering cross section from g′ to g. $R_{\ell m}$ is the real spherical harmonics function and $\phi_{\ell m}$ represent the flux moment. STRAUM solves Eq.(2) after spatially discretizing it using the LDEM-SCB [2] or DFEM methods [1] on unstructured tetrahedral meshes.

The spatially discretized form of Eq.(1) can be represented in an operator form as follows:

$$\mathrm{L}\psi = (\mathrm{MS}\phi + \mathrm{q}), \tag{2}$$

where the operator M is the moment-to-discrete operator, S is the scattering operator, ϕ represents the vector comprised of the moments, and L represent the transport operator comprised of streaming and collision terms.

Then, Eq.(2) can be expressed by using the discrete-to-moment operator D as follows :

$$(\mathrm{I} - \mathrm{DL}^{-1}\mathrm{MS})\phi = \mathrm{DL}^{-1}q \quad , \tag{3}$$

where L^{-1} represents the sweeping operator which updates the outgoing angular flux by using incoming fluxes over all the meshes for a given direction.

By defining operator A as $(\mathrm{I} - \mathrm{DL}^{-1}\mathrm{MS})$ and operator b as $\mathrm{DL}^{-1}q$, respectively, Eq.(3) can be represented as follows:

$$\mathrm{A}\phi = \mathrm{b}\,. \tag{4}$$

This linear equation is solved using the Krylov subspace methods in STRAUM. However, it should be noted in this formulation that the operator A is involved with the transport sweepings represented L^{-1} for all the directions.

2.2 Parallelization for Angular Directions

As mentioned above, in Eq.(3), L^{-1} operator means the transport sweepings which are performed in the independently pre-determined sweeping orders for each direction. Therefore, it is very natural that the sweeping is parallelized in angular directions. Spatial domain decomposition is also possible but it requires much more effort to reallocation of the tetrahedral meshes in each coarse spatial domain. So, as a first step, it is determined to parallelize the angular directions in STRAUM. In particular, the parallelization is applied in each octant because the directions in one octant are swept after the ones in the other octant for general problems having

reflective conditions. We used the OpenMP library based on shared memory to prevent the drastic increase of the memory in a single node. The parallelization on the angular directions was implemented using the OpenMP built-in construct 'Atomic' and clause 'Reduction'. The Atomic is a construct that allows each thread to access atomically a specific shared memory location, rather than exposing it to the possibility of multiple, simultaneous reading and writing threads. On the other hand, the Reduction clause is a data-sharing attribute clause that can be used to make thread-specific private variables that are subject to the operation specified by reduction at the end of the parallel region.

2.3 Result and Discussion

The parallel performance test is conducted on the quarter reactor pressure vessel problem, which was designed to test the STRAUM code convergence [4]. This problem is a practical problem on the nuclear reactor scale and it consists of a total 177,575 tetrahedral meshes as shown in Fig. 1. Figs. 2 shows the layout of RPV problem. The fuel assembly region was full of ZIRLO. The structure regions of baffle, barrel and vessel were delineated as SS304. The vacuum boundary condition was applied to the outer surfaces, and the reflective boundary condition was applied to the rest of the inner surface.

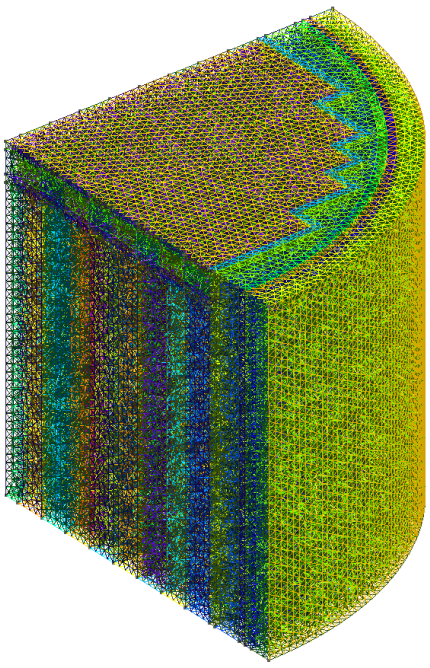

Fig. 1. Tetrahedron mesh for RPV problem [4]

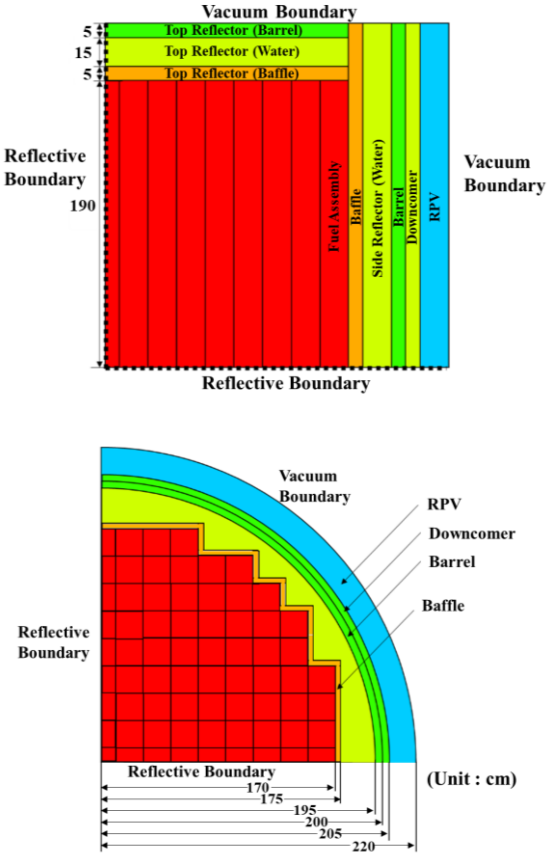

Fig. 2. Layout of RPV problem [4]

To test the parallelization of STRAUM on angular directions, the Gauss-Chebyshev angular quadrature of 6 azimuthal and 6 polar directions per octant was used to describe the direction of particle moving. Table I shows the calculation conditions for this problem. Table II shows the specification of the computer system used for this test. As shown in Table I, the P_3 scattering anisotropy was used and the GMRES option as the Krylov subspace method was used. Two different cases were tested depending onwhether the thermal upscatteirng is considered or not. In particular, the thermal upscatterings exist in thermal energy groups from 47 to 42 groups, which makes the convergence very slow.

Table I: Calculation conditions for RPV problem

Parameter	Contents
Number of tetrahedrons	177,575
Quadrature set	6 polar x 6 azimuthal Gauss-Chebyshev Quadrature per Octant
Anisotropic order	P_3
Thermal upscattering	From 42 to 47 / None

Krylov subspace method	GMRES
Convergence criteria	Ratio of L_2 norm of residual to $\|\vec{b}\|_2$ < 10^{-10}
DSA convergence criteria	Ratio of L_2 norm of residual to $\|\vec{b}\|_2$ < 10^{-5}

Table II: Specification of the workstation

Component	Specification
CPU	AMD Ryzen Thread ripper 3970X 32-Core (64 threads) Processor 3.69 GHZ
Compiler	Visual studio 2022 Intel DPC 2022++

Before the parallel test, we first checked the elapsed times for one sweep operation when DSA preconditioner for the single group chunk with a single thread is used in order to analyze the portions of the elapsed computing time for each calculation part. Table III summarizes the result of the analysis of the computing time. In Table III, the elapsed time of $\vec{b}$ setup is to construct the $\vec{b}$ vector at Eq.(4).

Table III: Elapsed time of single group chunk

Section	Time (sec)	Portion (%)
Sweep	639	88.2
$\vec{b}$ setup	79	10.9
DSA	6	0.8
All remaining part	1	0.1

The elapsed time for the sweeping calculations and $\vec{b}$ setup took 639 and 79 seconds, respectively, which correspond to 88.2 % and 10.9 % of the total elapsed time, while the elapsed time for DSA was only small portion. Figs. 3 and 4 show the speedup and parallel efficiency (%) of the parallelization with different number of threads relative to the computation with a single thread. It should be noted that this analysis of computing efficiency is only for sweeping calculation for a single energy group. As shown in the figures, the parallel efficiency decreases as number of threads and the 'Atomic' option shows better performances than the 'Reduction' option. For the 18 and 36 threads cases, the 'Atomic' option showed 75.6 % and 62.8 %, respectively while the 'reduction' option showed 60.4 % and 36.0 %, respectively.

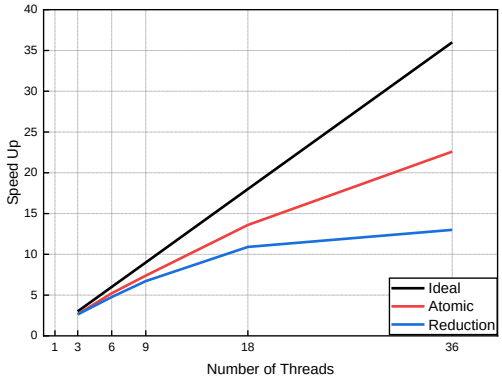

Fig. 3. Speed up with the number of threads

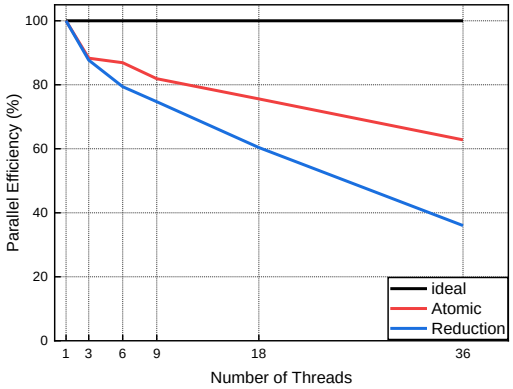

Fig. 4. Parallel efficiency with the number of threads

Fig. 5 and Fig. 6 compare the total elapsed time of this test problem and elapsed time of each section in STRAUM code, with and without the thermal upscattering, respectively. In Fig. 5, considering the thermal upscattering, the single thread calculation took about 22 hours to solve this test problem where most of the elapsed time is spent in sweeping calculations which took 19.6 hours. However, when using 36 threads, the total and sweeping calculations took 85 minutes and 71.5 minutes, which corresponds to 1/15.5 and 1/16.6 reductions, respectively, relatively to the single thread calculations. In Fig. 6, without considering the thermal upscattering, the single thread took about 13.4 hours to solve this test problem and sweeping calculations took 12.3 hours. However, when using 36 threads, the total and sweeping calculations took 46 minutes and 39 minutes, which corresponds to 1/17.5 and 1/18.9 reductions, respectively, relatively to the single thread calculations..

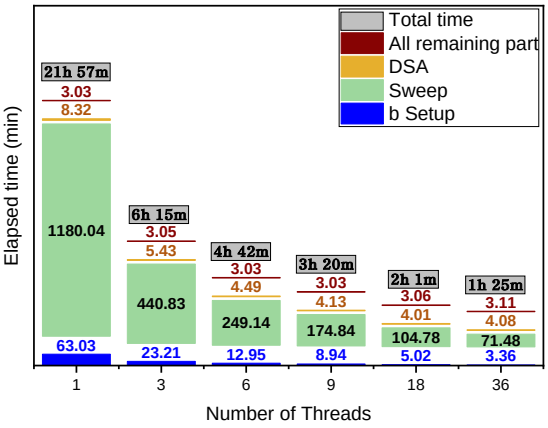

Fig. 5. Total elapsed time of RPV problem with the thermal upscattering

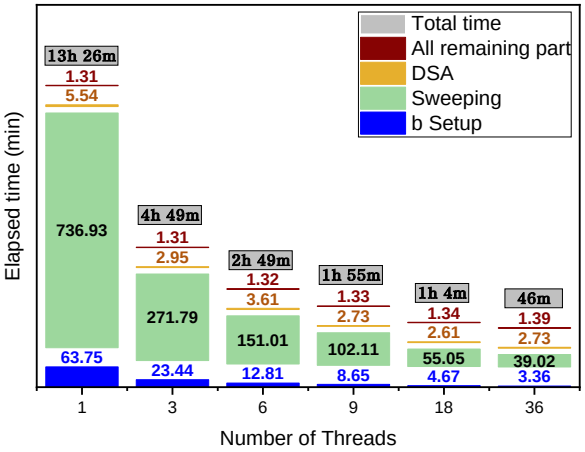

Fig. 6. Total elapsed time of RPV problem without the thermal upscattering

Fig. 7 shows the parallel efficiency of this problem as number of threads. The parallel efficiencies considering the thermal upscattering, for 18 and 36 threads is 60.0 % and 42.5 %, respectively, while the ones without considering the thermal upscattering, for 18 and 36 threads is 70.0% and 48.2 %, respectively.

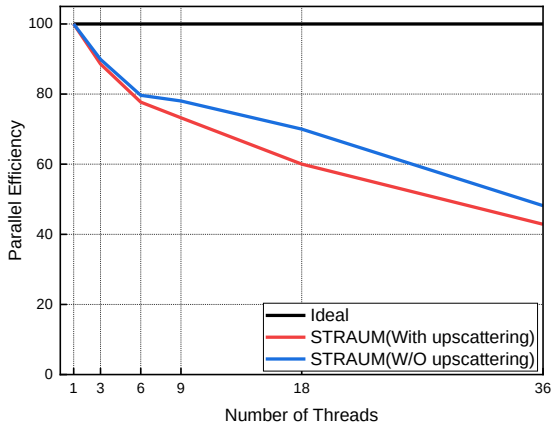

Fig. 7. Parallel efficiency of STRAUM

3. Conclusions

In this work, we presented the parallelization on the angular directions using OpenMP in the STRAUM code to reduce computing time. The analysis of the portions in computing time by different parts showed most of the computing time is taken in the sweeping calculation and the parallelization on angular directions will be efficient in reducing the sweeping calculations because the sweeping calculations in different directions for an octant are independent each other. The test of the parallelization was conducted by the quarter reactor pressure vessel problem which was discretized using 177,575 tetrahedral meshes, six azimuthal and six polar directions per octant, and P_3 scattering anisotropy.

From the results of the analysis, it was shown that the 'Atomic' option in OpenMP can be effectively used without memory increase. For the quarter reactor pressure vessel problem, the parallel efficiencies of 'Atomic' option was 75.6 % and 62.8 % for 18 and 36 threads, respectively, for only sweeping. For this test problem with the thermal upscattering, the total computing times using 18 and 36 threads was reduced by the factors of 15.5 and 16.6, respectively. Without the thermal upscattering, the total computing times using 18 and 36 threads were reduced by the factors of 17.5 and 18.9, respectively.

4. Acknowledgment

This work was supported by the NRF (National Research Foundation of Korea) through Project No. NRF-2019M2D2A1A02057890.

REFERENCES

[1] Todd A. Wareing, et al, Discontinuous Finite Element Sn Methods on Three Dimensional Unstructured Grids, Nuclear Science and Engineering, Vol. 138, pp. 256-268, 2001.
[2] S.G. Hong, Two subcell balance methods for solving the multigroup discrete ordinates transport equation with tetrahedral meshes, Nuclear Science and Engineering, Vol. 173, pp. 101-117, 2013.
[3] T.D. Long, and S.G. Hong, Implementation and Verification of adjoint neutron transport calculation in MUST code, Transactions of the Korea Nuclear Society Virtual Spring Meeting, 2020.
[4] MyeongHyeon Woo and Ser Gi Hong, STRAUM-MATXST: A Code System for Multi-group Neutron-Gamma Coupled Transport Calculation with Unstrctured Tetrahedral Meshes, Nuclear Engineering and Technology, Vol. 7, 2002.
[5] Chandra R, Dagum L, Kohr D, Menon R, Maydan D, and McDonald J, Parallel programming in OpenMP, Morgan kaufmann, 2001.