Improvement of Parallel Algorithm for MATRA Code

Seong-Jin Kim*, Kyong-Won Seo, Hyouk Kwon, Dae-Hyun Hwang
Korea Atomic Energy Research Institute, Daejeon, Korea
*Corresponding author: sjkim2@kaeri.re.kr

1. Introduction

A subchannel analysis code MATRA [1] is under development at KAERI for the evaluation of thermal margin in various types of reactor cores as well as for the multi-physics calculations coupled with neutronics and/or fuel performance codes. The feasibility study to parallelize the MATRA code was conducted in KAERI early this year. As a result, a parallel algorithm for the MATRA code has been developed to decrease a considerably required computing time to solve a big-size problem such as a whole core pin-by-pin problem of a general PWR reactor and to improve an overall performance of the multi-physics coupling calculations [2]. It was shown that the performance of the MATRA code was greatly improved by implementing the parallel algorithm using MPI communication. For problems of a 1/8 core and whole core for SMART reactor, a speedup was evaluated as about 10 when the numbers of used processor were 25. However, it was also shown that the performance deteriorated as the axial node number increased. In this paper, the procedure of a communication between processors is optimized to improve the previous parallel algorithm.

2. Methods and Results

In this section, the algorithm used to embody the parallel computing function into the MATRA code is briefly described. The improvement to optimize the previous parallel algorithm is also presented.

2.1 Description of Previous Parallel Algorithm

One of the specifications of the parallel algorithm for the MATRA code is to minimize modifying a structure of the previous MATRA code. Moreover, the parallel algorithm of the MATRA code can be summarized as axially dividing a whole of an axial domain to be interested into several parts. An each processor charges an each parts to be divided. The calculation result are transferred with the MPI communication and updated to the next outer iteration. A procedure to communicate between processors is depicted in Fig. 1. In the figure, the number in a square means a rank number of processors and a blue-colored arrow means a data flow. As shown in Fig.1, a master processor, RANK#0, handles all communications between processors and controls. An each processor sends the calculated result to the master processor, immediately after performing the work assigned to it. The master processor receives the updated data from all processors in order. After receiving all data, the master processor broadcasts them to other processors to be updated and the communication is synchronized. The next outer iteration is started and this procedure is repeated until a convergence criterion is satisfied.

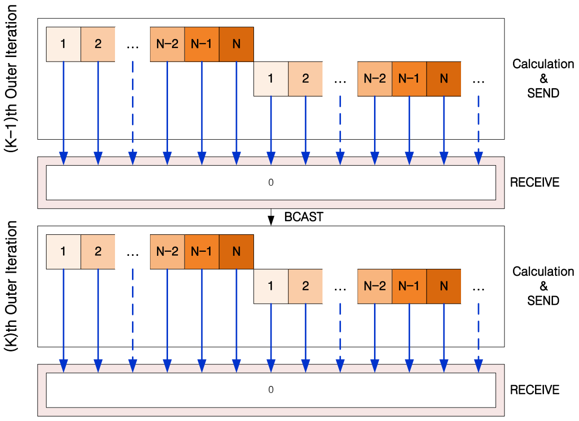

Fig. 1. In the previous parallel algorithm, the master processor gathers the calculated data from all processors and broadcasts updated date to all processors.

2.2 Improved Parallel Algorithm

In the previous parallel algorithm, the master processor receives data from all processor except for itself in order regardless of the processor number. That means the communication time is accumulated in the master processor and it can be affected on the performance. Thus, the period and counterpart for a communication were rearranged as shown in Fig. 2. First of all, the period of calculation and communication was divided into two parts. It means that the data communication comes after all of assigned calculations are finished, while data was sent to the master processor as often as one of assigned calculations in the previous parallel algorithm. Also, an amount of communication from each processor to the master processors was slashed. All of the calculated data except for the necessary imperative are directly transferred into neighbor processor. It means that the Nth processor communicates with (N+1)th processor and the last ranked processor communicates with the 1st ranked processor. In the parallel algorithm of the MATRA code, the values at the previous axial-node are needed to calculate the present axial-node and the present and these updated values are not used until the next outer iteration. Thus, the calculation and communication can be spilt.

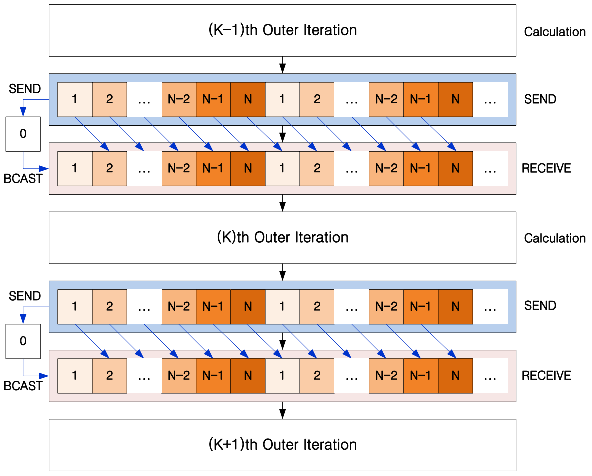

Fig. 2. The calculation and communication parts are spilt to optimize the communication time between processors and the communication method focused on the master node are change to be shared with others.

2.3 Result of Performance Test

As embodying the presented parallel algorithm, the performance test of a 1/8 and whole core for SMART reactor was conducted, when the axial node number is respectively 44 and 89. The channel numbers of each subchannel model are 2,331 and 16,780, and the rod numbers are 2,119 and 15,048, respectively. The gap numbers of each model are 4,536 and 33,252. To test the performance, 1 main-node and 4 sub-nodes were used. The main node has a specification of 2533 MHz, 8 CPUs, 24 GB RAM. The specifications of the subnodes are 2660 MHz, 12 CPUs, and 96 GB RAM. The communication between nodes uses the InfiniBand network. A wall-time was recorded to measure the speedup of the MATRA code as increasing the number of processors. The results of the performance test are depicted in Figs. 3 and 4. In the figures, open and closed symbol means the performance result of the previous and the present parallel algorithm, respectively. In the previous parallel algorithm, the performance deterioration as increasing the axial node number clearly appeared in case of 1/8 core model. In case of whole core model, decrease of the speedup was shown. On the other hand, the performance deterioration was not shown as the axial node number increased when the presented parallel algorithm in this paper. The speedup was rather to be increased in case of whole core problem as increasing the axial node number.

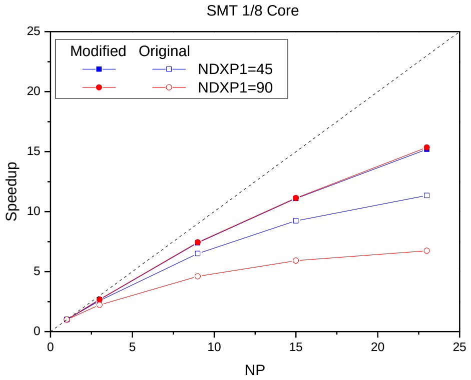

Fig. 3. This is the result of 1/8 core for SMART reactor, which shows that the performance of the present parallel algorithm is improved and stable regardless of the axial node number.

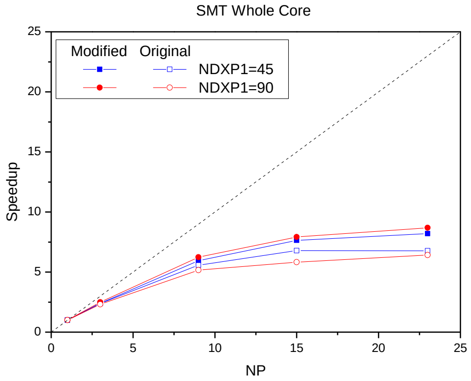

Fig. 4. This is the result of whole core for SMART reactor, which shows that the performance of the present parallel algorithm is improved and stable regardless of the axial node number.

3. Conclusions

To improve the performance deterioration of the parallelized MATRA code, the communication algorithm between processors was newly presented. It was shown that the speedup was improved and stable regardless of the axial node number. It is concluded that the presented communication algorithm was successful.

ACKNOWLEGEMENTS

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (No. 2012M2A8A4026261)

REFERENCES

[1] S.J. Kim, at al., Development and Assessment of Core T/H Code's Real Time Model for SMART Simulator, KAERI/TR-4904/2013, Korea, Jan., 2013.
[2] S.J. KIM, et al., "Parallelization of Subchannel Analysis Code MATRA," Transactions of the Korean Nuclear Society Spring Meeting, Jeju, Korea, May., 29-30, 2014.