PERFORMANCE OF A FINE-GRAINED PARALLEL MODEL FOR MULTI-GROUP NODAL-TRANSPORT CALCULATIONS IN THREE-DIMENSIONAL PIN-BY-PIN REACTOR GEOMETRY

Masahiro Tatsumi and **Akio Yamamoto**
Nuclear Fuel Industries, Ltd., Japan

Abstract

A production code SCOPE2 was developed based on the fine-grained parallel algorithm by the red/black iterative method targeting parallel computing environments such as a PC-cluster. It can perform a depletion calculation in a few hours using a PC-cluster with the model based on a 9-group nodal-SP3 transport method in 3-dimensional pin-by-pin geometry for in-core fuel management of commercial PWRs.

The present algorithm guarantees the identical convergence process as that in serial execution, which is very important from the viewpoint of quality management. The fine-mesh geometry is constructed by hierarchal decomposition with introduction of intermediate management layer as a block that is a quarter piece of a fuel assembly in radial direction. A combination of a mesh division scheme forcing even meshes on each edge and a latency-hidden communication algorithm provided simplicity and efficiency to message passing to enhance parallel performance. Inter-processor communication and parallel I/O access were realized using the MPI functions.

Parallel performance was measured for depletion calculations by the 9-group nodal-SP3 transport method in 3-dimensional pin-by-pin geometry with 340x340x26 meshes for full core geometry and 170x170x26 for quarter core geometry. A PC cluster that consists of 24 Pentium-4 processors connected by the Fast Ethernet was used for the performance measurement. Calculations in full core geometry gave better speedups compared to those in quarter core geometry because of larger granularity. Fine-mesh sweep and feedback calculation parts gave almost perfect scalability since granularity is large enough, while 1-group coarse-mesh diffusion acceleration gave only around 80%. The speedup and parallel efficiency for total computation time were 22.6 and 94%, respectively, for the calculation in full core geometry with 24 processors.

Introduction

Improvement of computing power makes an innovative computational method usable and useful in real applications. One example is the method of characteristics (MOC) in neutron transport problems. The MOC was firstly adopted in solving transport problem by Vladimirov in the late 1950s[1] but it was limited to simple geometry. Forty years later, on a much faster computer, the MOC was applicable to whole-core geometry[2-5]. Another example is the advance nodal method (ANM) [6,7]. In the late 1980s or the early 1990s, a core calculation code based on the ANM required several hours for a tracking calculation of a cycle of PWR. Nowadays it is possible to perform such a calculation in a few minutes using the latest but commodity computing hardware such as Pentium-4 based PCs. Successful application of ANM in the 3-dimensional system brought more reliability to in-core fuel management. This method would be one of major methods to in the field of in-core fuel management. It may be used as a screening method in the loading pattern optimization as we enjoy more powerful PCs. In this way, fast and reliable simulators based on the ANM will have been very useful for in-core fuel management.

Powerful computing environment provide the other direction of development: the best-estimate core calculation code. Improvement of computing power makes it possible to introduce more precise and less approximated calculation methods. This can be realized by improving resolution for spatial, energy, angular domains in computational methods. Firstly, 3-dimensional pin-by-pin calculation gives good spatial resolution without uncertainty due to reconstruction of pin power[8] in the ANM with homogeneous nodes. Secondary, multi-group calculation instead of the conventional 2-group calculation can provide more proper treatment of spectrum mismatch between different types of fuel assemblies. Thirdly, transport theory method gives more accurate results in the regions where gradient of neutron flux is very steep compared to the conventional diffusion theory method.

SCOPE2 is the very code in the second direction as a production code for in-core fuel management[9,10]. It solves the multi-group transport equations based on the simplified P3 approximation[11] within the framework of the advanced nodal method in 3-dimensional pin-by-pin geometry. All the features needed for in-core fuel management are fully implemented such as critical boron or k-effective search, burnup calculation, neutron-thermal feedback, RCC cusping and grid models, restart/reload calculation, geometry transform, and extensive editing. Different homogenized cross sections are assigned to each fine mesh, in which the effect due to homogenization is taken into account by user of the SPH[12] factors. The nodal expansion method (NEM) based on the diffusion method is applied in the axial direction in order to treat deviation of neutron flux. In the models for RCC cusping and grid treatment, re-homogenization is applied using flux distribution in the intra-nodal NEM solution. Therefore pin-by-pin or pellet-by-pellet power distribution can be estimated with good accuracy.

Since SCOPE2 incorporates the very precise model, computational load could be very large. In order to reduce computation time, we derived and implemented an efficient formulation of response matrix without nonlinear iteration scheme for fine-mesh solutions and a multi-grid acceleration method based on one-group

diffusion method[10]. Those greatly contribute reduction of computation time. Parallel computing is also adopted in the last resort; SCOPE2 was designed and implemented for parallel computing environment. In order to guarantee the identical convergence process for all the processor configurations, a fine-grained parallel algorithm are implemented.

In this paper, the parallel model in SCOPE2 and its performance will be discussed.

Parallel Model

Geometry Management

The parallel model in SCOPE2 can be described as a parallel model based on fine-grained Red/Black sweep algorithm with 2-diemnsional domain decomposition. This algorithm can guarantee the identical convergence process as that in serial execution, but it might degrade performance because of frequent communication between processors within the most-inner iterations. Through development of SCOPE in the former study[13], however, it was found that non-blocking message passing helped good performance even for such a situation. Furthermore, the following techniques were developed and implemented in SCOPE2.

- Construction of fine-mesh geometry with hierarchical decomposition
- Red/Black sweep algorithm forcing even-mesh division
- Latency-hidden communication algorithm based on non-blocking message passing

Hereafter, the details of these topics will be reviewed.

Construction of Fine-Mesh Geometry with Hierarchical Decomposition

The first technique concerns the method of mesh division and how to handle them. A simple way to handle 3-dimensional flux distribution can be shown in Fig. 1(a). This is a straightforward, however, it has a server defect to apply to parallel computing from the viewpoints of memory management. It is quite troublesome to handle irregularly decomposed geometry on each processor in order to minimize memory usage. On the other hand, hierarchical decomposition is a smart way to handle such geometry. Figure 1(b) indicates a different way of dimensioning by changing the meaning of the last index. In this way, 3-dimensional fine-mesh geometry can be constructed as an aggregation of blocks in which fine meshes are assigned as shown in Fig. 2. This dimensional management enables efficient array structure on the memory; total memory usage is not dependent on the number of processors, and the cache effect can be expected by contiguous memory access especially when large number of processors are used to have small fraction of memory space assigned to each processor.

```
dimension  flux(kg, x, y, z)
```

(a) Straightforward way

```
dimension  flux(kg, x, y, block)
```

(b) Hierarchical way

Fig. 1 Comparison of dimensioning scheme for flux distribution in 3-diemensional systems

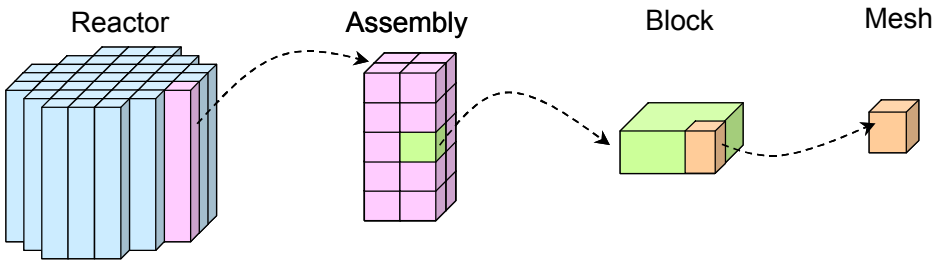


Fig.2 Hierarchy of calculation geometry in 3-D fine-mesh system

Domain decomposition in parallel execution is done by distributing blocks to each processor participating in the calculation. In SCOPE2, 2-dimensional domain decomposition is done instead of 3-dimensional one because there is dependence on calculation orders in the axial direction in thermal feedback calculation where a sweep is directional from the bottom to the top. In order to achieve high efficiency, the decomposition is done based on the following rules.

- Computation load should be equivalent on each processor.
- Surface area on the processor boundary should be minimized.
- The number of processors to be connected each other should be minimized and equalized.

As a solution to the constraints, in SCOPE2, a system is decomposed to shortcake-style volumes. Since blocks inside fuel assemblies and reflectors have different computing load, the ratio of two block types should be constant among processors. For example, Fig. 3 shows a decomposition scheme using 16 processors for a 3-loop type PWR core. Thick lines in Fig.3 indicate the boundary of fuel assemblies. When some nodes have different computing power, it is possible to adequately decompose the domain by distributing the loads proportionally to computing power for the node.

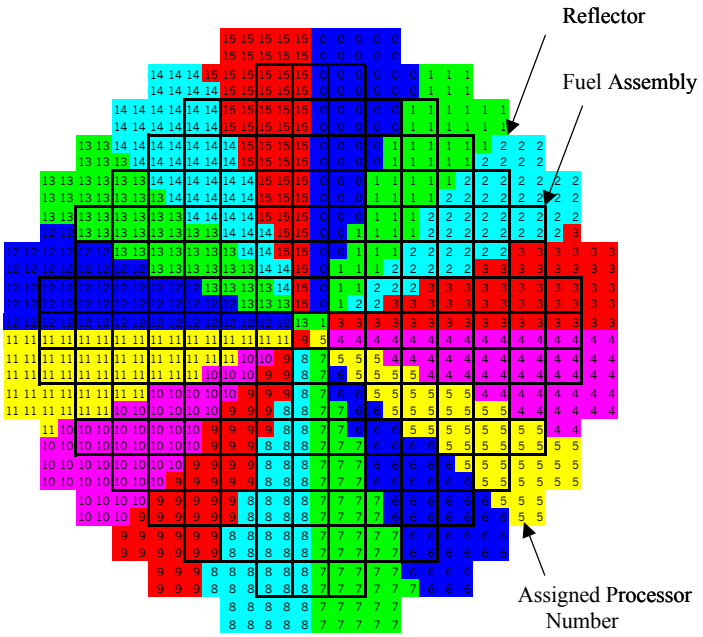


Fig.3 Example of decomposition scheme using 16 processors in 3-loop type PWR core

Red/Black Sweep Algorithm Forcing Even-Mesh Division

The neutron transport equation is solved by the Red/Black iterative method in fine mesh system within a block. In SCOPE2, the number of mesh division in a block is forced to be even for radial direction (i.e. x and y axes) in order to enhance the performance. With the even mesh number, the coloring scheme can be fixed inside blocks, which leads to symmetric data exchange between adjacent blocks and a simple implementation. Figure 4 illustrates a typical division scheme for a Westinghouse-type fuel assembly with 4 blocks in which 10x10 meshes are assigned. More than one mesh is assigned for the fuel pins at the row and columns of peripheral and central positions. One reason for this is to give good accuracy at the interface between assemblies so that steep gradient of flux distribution can be treated more accurately where spectral mismatch is observed. The other reason is for even-mesh division. In this case, the mesh in the upper-left corner of a block is always colored as black. The number of red or black meshes on a edge of a block is 5 for both color, which leads to symmetric data exchange. This is important in message passing communication.

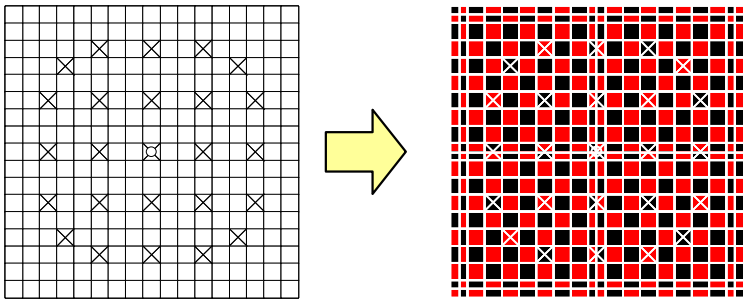


Fig. 4 Typical division scheme in Even-Mesh Red/Black Method for the Westinghouse–type fuel assembly

Latency-Hidden Communication Algorithm Based on Non-Blocking Message Passing

Another advantage of the even-mesh division is that calculation can be done in arbitrary order of blocks as far as it is not on a boundary among processors. This characteristic enables the latency-hidden communications by overlapping computation and communication; it firstly initiates non-blocking communications to update boundary conditions among processors, then calculations are performed for blocks that are not on the processor boundary followed by calculations for those on the boundary.

Fine-Grained Parallel Algorithm

A basic concept for a fine-grained parallel algorithm is to follow the same calculation orders in the serial execution. This is very important from the viewpoint of quality management. A simple flow diagram in the calculation kernel of SCOPE2 is shown in Fig. 5. For example, exchange of partial neutron currents between blocks done via temporary memory space in serial execution is performed via network communication in parallel execution on PC clusters. Integration over the whole geometry or over all energy groups in parallel execution is equivalent to that in serial execution when an integrated variable has enough precision (i.e. double precision or more).

The block is a unit for domain decomposition and coarse mesh acceleration. The domain decomposition is done for in 2-D manner as mentioned before. Each processor has a fraction of whole geometry, therefore all the functional parts in SCOPE2 are fully parallelized for initialization, fine-mesh calculation, 1-group coarse mesh diffusion acceleration, summary edit, feedback calculation, cross section update. File access to restart files is also parallelized by MPI-IO from the viewpoint of performance and handiness. The parallel I/O access is done by the so-called level-3 access[14] using collective, noncontiguous requests for irregularly distributed arrays using derived data types.

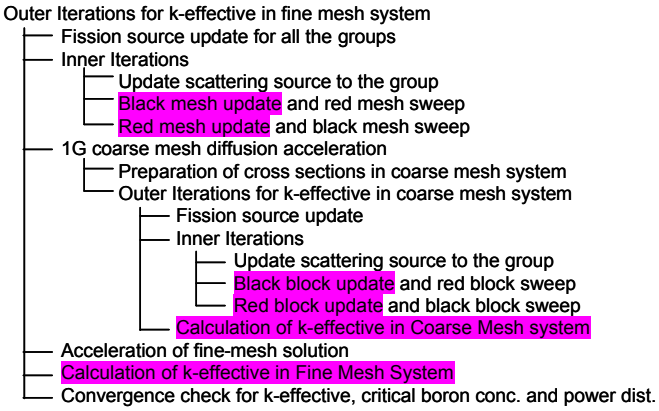


Fig. 5 Fine-grained Parallel Algorithm in the Calculation Kernel
(Intra-processor communications are preformed in the marked sections)

Performance

Geometry and Cluster

The parallel performance of SCOPE2 calculations was measured in a typical design calculation of a cycle of 3-loop type PWR core. A sequence of depletion calculations with 22 steady states (12 burnup points by the predictor/collector method) is performed to obtain properties such as critical boron concentration and pin power distribution at the each state by changing the number of processors participating in the calculations. Note that numerical results were identical among all the cases by the fin-grained parallel algorithm. In order to obtain the performance in small and large geometry configurations, two sets of measurement were performed with the meshing scheme given in Table 1. A 2-dimensional geometry model in quarter core symmetry is shown in Fig. 6

Table 1 Parameters for performance measurement in production calculations

Core Symmetry	Total Mesh Numbers	Total Memory	Processors
Quarter Core	170 x 170 x 260	Approx. 1.2GB	1 – 16
Full Core	340 x 340 x 260	Approx. 5GB	6 – 24

A PC-cluster used to measure the parallel performance consists two kinds of processors with different CPU clocks. Figure 7 shows a block diagram of the cluster and it indicates that two island of sub-cluster is connected via Gigabit Ethernet links. In order to measure the performance consistently for the configuration with more than 16 processors, the capability of multiform distribution of workloads was not used so that it

could simulate a homogeneous cluster.

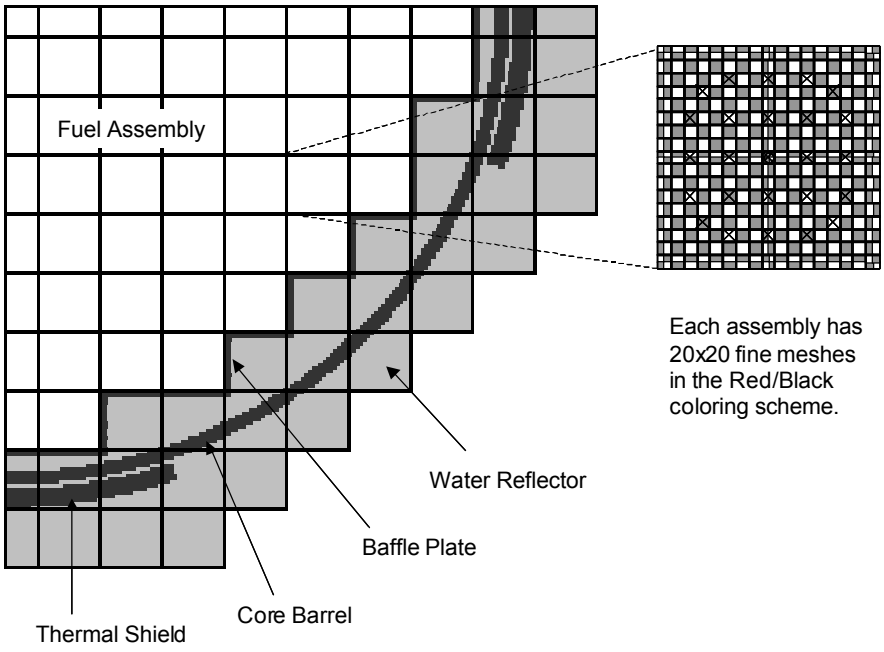


Fig. 6 Radial cross section view of calculation system for quarter core.

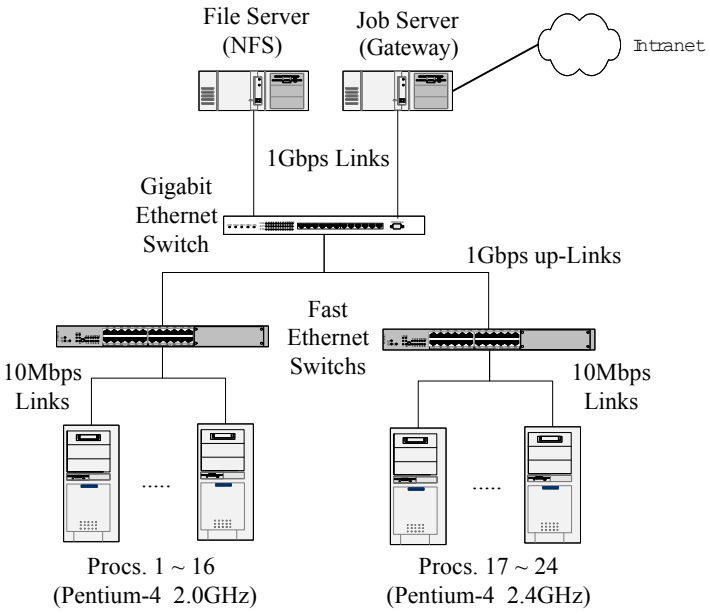
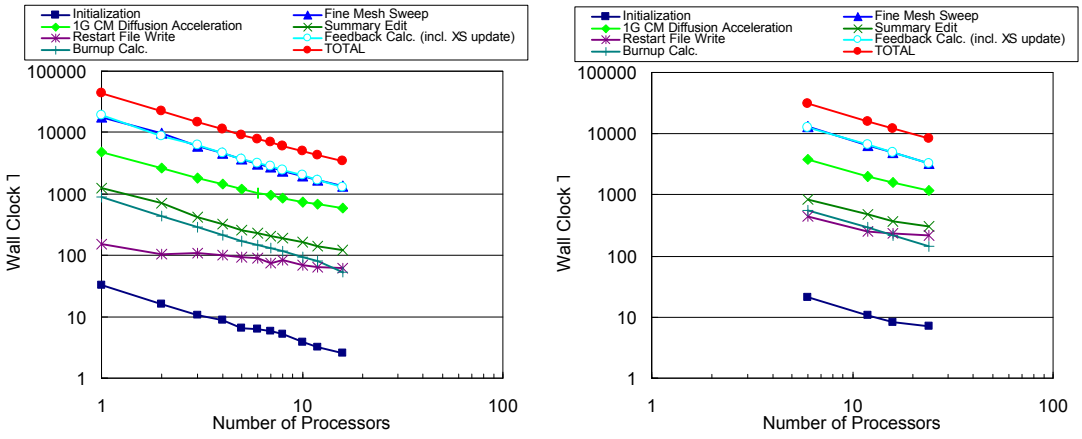


Fig. 7 Network diagram of a PC cluster used for measurement of parallel performance

Results

Figure 8(a) and 8(b) show elapsed time for each part of the design calculations in quarter and full core geometry, respectively. Table 2 summarizes the time profile for plotting Fig. 6. Parallel efficiency in each part is given in Table 3. The linearity is quite good for total computation time versus the number of processors; parallel efficiencies of 80% and 94% were obtained for quarter and full core geometry systems, respectively. The efficiency is reasonable and fairly acceptable since the number of processors to be used ranges from 16 to 24 in actual design calculations.



(a) Quarter core geometry (b) Full core geometry

Fig. 6 Elapsed time in each calculation part versus the numbers of processors

Table 2 Profile of elapsed time by each part in parallel execution (unit: sec)

Geometry	# of Procs.	Initalize	Fine Mesh Sweep	1G CM Diff. Accel.	Summary Edit	Restart File Write	FB Calc. (incl. XS update)	Burnup Calc.	TOTAL
Quarter Core	1	32	17705	4882	1236	154	18923	873	43807
	2	16	9954	2639	701	104	8622	432	22466
	3	11	6013	1819	418	109	6295	287	14951
	4	9	4606	1428	318	101	4671	214	11346
	5	7	3706	1187	261	94	3719	169	9143
	6	6	3086	1049	228	91	3141	146	7748
	7	6	2704	942	209	76	2830	133	6900
	8	5	2357	869	188	83	2458	117	6076
	10	4	1955	748	162	70	2025	94	5056
	12	3	1656	675	140	65	1692	80	4310
	16	3	1419	583	122	62	1183	54	3425
Full Core	6	21	13047	3805	852	452	12701	562	31440
	12	11	6427	1968	479	248	6540	294	15966
	16	8	4932	1623	374	234	4851	216	12239
	24	7	3285	1166	304	216	3230	145	8354

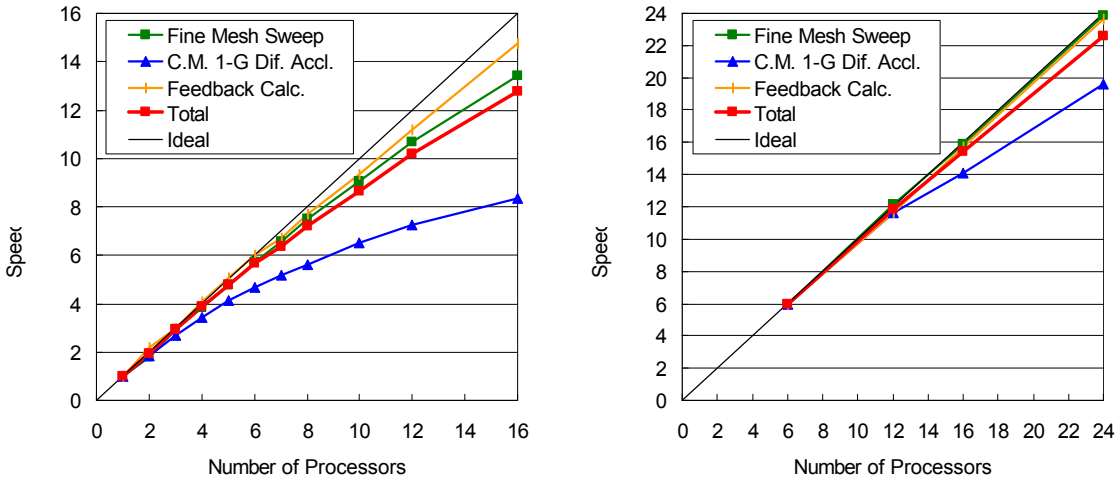
Table 3 Speedup results for each part in parallel execution and total parallel efficiency

Geometry	# of Procs.	Initalize	Fine Mesh Sweep	1G CM Diff. Accel.	Summary Edit	Restart File Write	FB Calc. (incl. XS update)	Burnup Calc.	TOTAL	Parallel Efficiency
Quarter Core	1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	100%
	2	2.0	1.8	1.9	1.8	1.5	2.2	2.0	1.9	97%
	3	3.0	2.9	2.7	3.0	1.4	3.0	3.0	2.9	98%
	4	3.7	3.8	3.4	3.9	1.5	4.1	4.1	3.9	97%
	5	4.9	4.8	4.1	4.7	1.6	5.1	5.2	4.8	96%
	6	5.0	5.7	4.7	5.4	1.7	6.0	6.0	5.7	94%
	7	5.5	6.5	5.2	5.9	2.0	6.7	6.6	6.3	91%
	8	6.1	7.5	5.6	6.6	1.9	7.7	7.5	7.2	90%
	10	8.5	9.1	6.5	7.7	2.2	9.3	9.3	8.7	87%
	12	10.1	10.7	7.2	8.8	2.4	11.2	11.0	10.2	85%
	16	12.4	13.4	8.4	10.1	2.5	14.7	16.3	12.8	80%
Full Core	6	6.0	6.0	6.0	6.0	6.0	6.0	6.0	6.0	100%
	12	11.6	12.2	11.6	10.7	10.9	11.7	11.5	11.8	98%
	16	15.5	15.9	14.1	13.7	11.6	15.7	15.6	15.4	96%
	24	17.4	23.8	19.6	16.8	12.5	23.6	23.2	22.6	94%

Approximately 80% of the computation time is consumed in fine-mesh sweep and cross section update due to feedbacks by thermal condition and estimation of boron concentrations. The fine-mesh sweep scored almost perfect scalability in full core geometry as shown in Fig. 7 despite the fine-grained parallel algorithm. The speedup factor of 23.8 for 24 processors was measured in full core geometry as given in Table 3. The reason why good scalability was obtained was that the computing cost was much greater than communication cost in such heavy calculations in fine-mesh geometry. The synthesis of cross section is also a heavy process. Furthermore the update of cross sections can be done independently in each processor without message passing to give perfect scalability, which is indicated by “Feedback Calc.” in Fig. 7.

On the other hand, scalability of 1-group coarse-mesh diffusion acceleration, a third time-consuming part, is not so good because of lack of computation load. The speedup for the acceleration degrades rapidly as the number of processors increase to give only 82% of parallel efficiency for the calculation in full core geometry with 24 processors. It was because the granularity, the ratio of computation to communication, was much smaller than that in fine-mesh calculations. As seen in Fig.7, scalability for total computation time is not so affected by the degradation since the ratio of the acceleration to total time is small: approximately 10 to 15 percent. Therefore practical and effective speedup can be expected for middle-class PC clusters with 20 or 30 processors. For large-scale clusters, good scalability can be expected provided the geometry is divided into finer blocks, i.e. meshes, along z-axis such as 340x340x60 or 340x340x120.

The time for restart file writing was also reduced in parallel execution although the file was written over the network file system (NFS). Restart data of 4116MB were stored in total in 216 seconds when 24 processors were used in full core geometry, which resulted in about 19MB/sec of the bandwidth for writing. This was almost equivalent to the bandwidth for writing to a physical disk array. Therefore the bottleneck was assumed to be in the both of file access over NFS and that to the physical disk array system.



(a) Quarter core geometry (b) Full core geometry

Fig. 7 Speed-up curves in design calculation for a 3-loop PWR core

Conclusion

A fine-grained parallel algorithm was developed based on the red/black iterative method for the multi-group nodal-transport calculations in 3-dimensional pin-by-pin geometry. The present algorithm guarantees the identical convergence process in that in serial execution, which is very important from the viewpoint of quality management. The fine-mesh geometry is constructed by hierarchal decomposition with introduction of intermediate management layer as a block that is a quarter piece of a fuel assembly in radial direction. A combination of a mesh division scheme forcing even meshes on each edge and a latency-hidden communication algorithm provided a simplicity and efficiency message passing.

Parallel performance was measured for depletion calculations by the 9-group nodal-SP3 transport method in 3-dimensional pin-by-pin geometry of 3-loop type PWR core using a PC cluster that consists of 24 Pentium-4 processors connected each other by the Fast Ethernet. Calculations in full core geometry gave better speedup compared to those in quarter core geometry because of large granularity. Fine-mesh sweep and feedback calculation parts gave almost perfect scalability since granularity is large enough, while 1-group coarse-mesh diffusion acceleration gave only 80%. The speedup and parallel efficiency for total computation time were 22.6 and 94%, respectively, for the calculation in full core geometry with 24 processors.

It has been confirmed that a depletion calculation with very precise model based on the 9-group nodal-SP3 transport method in 3-diemnsional pin-by-pin geometry is almost scalable on the PC cluster of 24 processors and it can be completed within a few hours.

Reference

1. V. S. Vladimirov, Ph.D. Dissertation, V. A. Stecklova Mathematics Institute, USSR (1959)
2. S. Kosaka, "Transport Theory Calculation of Heterogeneous Multi-Assembly Problem by Characteristics Method with Direct Neutron Path Linking Technique," *J. Nucl. Sci. Technol*, **37**, 1015 (2000).
3. N. Z. Cho, *et al*., "Whole-Core Heterogeneous Transport Calculations and Their Comparisons with Diffusion Results," *Trans. Am. Nucl. Soc.*, **83**, 292 (2000).
4. K. S. Smith and J. D. Rhodes, III, "Full-Core, 2-D, LWR Core Calculations with CASMO-4E," *Proc. PHYSOR2002*, Seoul, Korea, Oct. 2002, p. 13A-04 (2002).
5. H. G. Joo, *et. al.*, "Dynamic Implementation of the Equivalence Theory in the Heterogeneous Whole Core Transport Calculation," *Proc. PHYSOR2002*, Seoul, Korea, Oct. 2002, p. 13A-02 (2002).
6. K. Koebke, "A New Approach to Homogenization and Group Condensation," IAEA Technical Committee Meeting on Homogenization Methods in Reactor Physics, IAEA-TECDOC-231, Lugano (1978).
7. K. S. Smith, "Assembly Homogenization Techniques for Light Water Reactor Analysis," *Prog. Nucl. Energy*, 17, 303 (1986).
8. K. R. Rempe and K. S. Smith, "SIMULATE-3 Pin Power Reconstruction: Methodology and Benchmarking," *Nucl. Sci. Eng.*, **103**, 334 (1989).
9. M. Tatsumi and A. Yamamoto, "SCOPE2: Object-Oriented Parallel Code for Multi-Group Diffusion/Transport Calculations in Three-Dimensional Fine-Mesh Reactor Core Geometry," *Proc. PHYSOR2002*, Seoul, Korea, Oct., 2002, p. 12A-01 (2002).
10. M. Tatsumi and A. Yamamoto, "Advanced PWR Core Calculation Based on Multi-group Nodal-transport Method in Three-dimensional Pin-by-Pin Geometry," *J. Nucl. Sci. Technol.*, **Vol. 40**, Number 6 (2002). [to be published]
11. E. M. Gelbard, "Application of Spherical Harmonics Method to Reactor Problems," WAPD-BT-20 (1960).
12. A. Hébert, "A Consistent Technique for the Pin-by-Pin Homogenization of a Pressurized Water Reactor Assembly," *Nucl. Sci. Eng.*, **113**, 227 (1993).
13. M. Tatsumi and A. Yamamoto, "Object-Oriented Three-Dimensional Fine-Mesh Transport Calculation on Parallel/Distributed Environments for Advanced Reactor Core Analyses," *Nucl. Sci. Eng.*, **141**, 190 (2002).
14. W. Gropp, E. Lusk and A. R. Thakur, *Using MPI-2: Advanced Features of the Message-Passing Interfac*e, MIT Press, Cambridge, Massachusetts (1994).