



RIST/TOKYO  
GeoFEM Report

Research Organization for  
Information Science & Technology

Kengo Nakajima

OpenMP/MPI Hybrid vs. Flat MPI on  
the Earth Simulator: Parallel Iterative  
Solvers for Finite Element Method

October 2003

RIST/TOKYO  
GeoFEM Report

Research Organization for  
Information Science & Technology

Kengo Nakajima

OpenMP/MPI Hybrid vs. Flat MPI on  
the Earth Simulator: Parallel Iterative  
Solvers for Finite Element Method

October 2003

<b>TITLE</b> OpenMP/MPI Hybrid vs. Flat MPI on the Earth Simulator: Parallel Iterative Solvers for Finite Element Method	
<b>AUTHORS</b> Kengo Nakajima (RIST)	
<b>CONTACT PERSON</b> Kengo Nakajima (nakajima@tokyo.rist.or.jp)	
<b>KEY WORDS AND PHRASES</b> Earth Simulator, SMP Cluster, Hybrid, Flat MPI, Unstructured Grid, GeoFEM	
<b>ABSTRACT</b> An efficient parallel iterative method for finite element method has been developed for symmetric multiprocessor (SMP) cluster architectures with vector processors such as the Earth Simulator. The method is based on a three-level hybrid parallel programming model, including message passing for inter-SMP node communication, loop directives by OpenMP for intra-SMP node parallelization and vectorization for each processing element (PE). Simple 3D linear elastic problems with more than $2.2 \times 10^9$ DOF have been solved using $3 \times 3$ block ICCG(0) method with additive Schwarz domain decomposition and PDJDS/CM-RCM reordering on 176 nodes of the Earth Simulator, achieving performance of 3.80 TFLOPS.	
<b>REPORT DATE</b> October 14th, 2003.	<b>TOTAL NO. OF PAGES</b> 12
<b>ANY OTHER IDENTIFYING INFORMATION OF THIS REPORT</b> to be presented at WOMPEI2003(International Workshop on OpenMP: Experiences and Implementations) /ISHPC-V (The fifth International Symposium on High Performance Computing), October 20-22, 2003, Tokyo, Japan. to be published as Lecture Notes in Computer Science (LNCS) 2858.	

Research Organization for Information Science & Technology (RIST)  
2-2-54, Nakameguro, Meguro-ku Tokyo, 153-0061, Japan  
TEL: +81-3-3712-5321 FAX: +81-3-3712-5552  
<http://geofem.tokyo.rist.or.jp/>

**Copyright©2003 Research Organization for Information Science & Technology (RIST)**

# OpenMP/MPI Hybrid vs. Flat MPI on the Earth Simulator: Parallel Iterative Solvers for Finite Element Method

Kengo Nakajima

Research Organization for Information Science and Technology (RIST)  
2-2-54 Naka-Meguro, Meguro-ku, Tokyo 153-0061, Japan  
nakajima@tokyo.rist.or.jp

**Abstract.** An efficient parallel iterative method for finite element method has been developed for symmetric multiprocessor (SMP) cluster architectures with vector processors such as the Earth Simulator. The method is based on a three-level hybrid parallel programming model, including message passing for inter-SMP node communication, loop directives by OpenMP for intra-SMP node parallelization and vectorization for each processing element (PE). Simple 3D linear elastic problems with more than  $2.2 \times 10^9$  DOF have been solved using  $3 \times 3$  block ICCG(0) method with additive Schwarz domain decomposition and PDJDS/CM-RCM reordering on 176 nodes of the Earth Simulator, achieving performance of 3.80 TFLOPS.

## 1 Introduction

### 1.1 SMP Cluster Architecture and Hybrid Parallel Programming Model

Recent technological advances have allowed increasing numbers of processors to have access to a single memory space in a cost-effective manner. As a result, symmetric multiprocessor (SMP) cluster architectures have become very popular as teraflop-scale parallel computers, such as the Accelerated Strategic Computing Initiative (ASCI, currently "Advanced Simulation and Computing (ASC)") [17] machines and the Earth Simulator [18].

In order to achieve minimal parallelization overhead, a multi-level hybrid programming model [2,3,6,12] is often employed for SMP cluster architectures. The aim of this method is to combine coarse-grain and fine-grain parallelism. Coarse-grain parallelism is achieved through domain decomposition by message passing among SMP nodes using a scheme such as Message Passing Interface (MPI) [20], and fine-grain parallelism is obtained by loop-level parallelism inside each SMP node by compiler-based thread parallelization such as OpenMP [21].

Another often used programming model is the single-level flat MPI model [2,3,6,12], in which separate single-threaded MPI processes are executed on each processing element (PE). The advantage of a hybrid programming model over flat MPI is that there is no message-passing overhead in each SMP node. This is achieved by allowing each thread to access data provided by other threads directly by accessing the shared memory instead of using message passing. However, a hybrid approach usually requires more complex programming.

Although a significant amount of research on this issue has been conducted in recent years [2,3,6,12], it remains unclear whether the performance gains of this hybrid approach compensate for the increased programming complexity. Many examples show that flat MPI is rather better [2,3,6,12], although the efficiency depends on hardware performance (CPU speed, communication bandwidth, memory bandwidth), features of applications, and problem size [13].

## 1.2 Overview

In 1997, the Science and Technology Agency of Japan (now, the Ministry of Education, Culture, Sports, Science and Technology, Japan) began a five-year project to develop a new super-computer, the Earth Simulator [18]. The goal of the project was to develop both hardware and software for earth science simulations. The Earth Simulator has SMP cluster architecture and consists of 640 SMP nodes, where each SMP node consists of 8 vector processors. The peak performance of each PE is 8 GFLOPS, and the overall performance of the system is 40 TFLOPS. Each SMP node of the Earth Simulator has 16 GB of memory, corresponding to 10 TB for the entire system. The present study was conducted as part of research toward developing a parallel finite-element platform for solid earth simulation, named GeoFEM [19].

In this paper, parallel preconditioned conjugate-gradient (CG) iterative solvers with incomplete Cholesky (IC) factorization have been developed for finite-element method using a three-level hybrid parallel programming model on the Earth Simulator. Individual PE of the Earth Simulator is a vector processor, therefore third-level of parallelism for vector processing should be considered in addition to the two levels, OpenMP and MPI. Following three levels of parallelism are considered:

- Inter-SMP node MPI for communication
- Intra-SMP node OpenMP for parallelization
- Individual PE compiler directives for vectorization

In flat MPI approach, communication among PEs through MPI and vectorization for individual PE have been considered for the Earth Simulator. In the hybrid parallel programming model, the entire domain is partitioned into distributed local data sets [7,9,10], and each partition is assigned to one SMP node. On the contrast, each partition corresponds to each PE in the flat MPI.

In order to achieve efficient parallel/vector computation for applications with unstructured grids, the following three issues are critical:

- Local operation and no global dependency
- Continuous memory access
- Sufficiently long innermost loops for vectorization

A special reordering technique proposed by Washio et. al. [16] has been integrated with parallel iterative solvers with localized preconditioning developed in the GeoFEM project [9,19] in order to attain local operation, no global dependency, continuous memory access and sufficiently innermost long loops.

In the following part of this paper, we give an overview of parallel iterative solvers in GeoFEM, special reordering techniques for parallel and vector computation on SMP nodes, and present the results for an application to 3D solid mechanics on the Earth Simulator. Developed hybrid parallel programming model has been compared with the flat MPI programming model. Finally, recent results of solid earth simulations are shown.

## 2 Parallel Iterative Solvers in GeoFEM

### 2.1 Distributed Data Structures

A proper definition of the layout of the distributed data structures is an important factor determining the efficiency of parallel computations with unstructured meshes. The local data struc-

tures in GeoFEM are node-based with overlapping elements, and as such are appropriate for the preconditioned iterative solvers used in GeoFEM (Fig.1) [9,19].

## 2.2 Localized Preconditioning

The incomplete lower-upper (ILU)/Cholesky (IC) factorization method is one of the most popular preconditioning techniques for accelerating the convergence of Krylov iterative methods. Factorization by backward/forward substitution is repeated in each iteration. This factorization requires global data dependency, and is not suitable for parallel processing in which the locality is of utmost importance. The localized ILU used in GeoFEM is a pseudo ILU preconditioning method that is

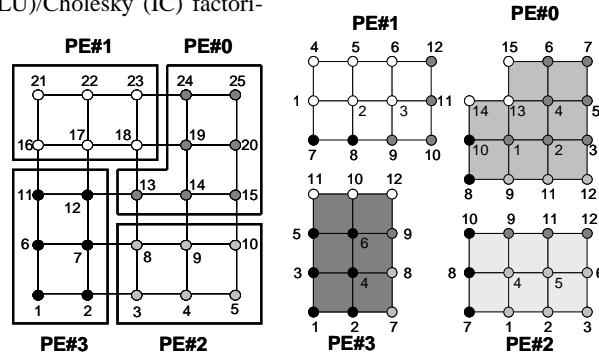


Fig. 1 Node-based partitioning into four PEs [9,19].

suitable for parallel processors. The ILU operation is performed locally for a coefficient matrix assembled on each processor by zeroing out components located outside the processor domain. This localized ILU provides data locality on each processor and good parallelization because no inter-processor communications occur during ILU operation. This idea is originally from the incomplete block Jacobi preconditioning method [1,15]. In order to stabilize localized ILU preconditioning, additive Schwarz domain decomposition (ASDD) for overlapped regions [15] has been introduced.

## 3 Reordering Methods for Parallel/Vector Performance on SMP Nodes

### 3.1 Cyclic Multicolor – Reverse Cuthill McKee Reordering

In order to achieve efficient parallel/vector computation for applications with unstructured grids, the following three issues are critical, (1) Local operations and no global dependency, (2) Continuous memory access and (3) Sufficiently long innermost loops for vector computation [10]. For unstructured grids, in which data and memory access patterns are very irregular, the reordering technique is very effective for achieving highly parallel and vector performance, especially for factorization operations in ILU/IC preconditioning.

The popular reordering methods are Reverse Cuthill-McKee (RCM) and multicoloring [14]. RCM method is a typical level set reordering method. In Cuthill-McKee reordering, the elements of a level set are traversed from the nodes of the lowest degree to those of the highest degree according to dependency relationships, where the degree refers to the number of nodes connected to each node. In RCM, permutation arrays obtained in Cuthill-McKee reordering are reversed. RCM results in much less fill-in for Gaussian elimination and is suitable for iterative methods with IC or ILU preconditioning. Multicoloring (MC) is much simpler than RCM. MC is based on an idea where no two adjacent nodes have the same color.

In both methods, elements located on the same color (or level set) are independent. Therefore, parallel operation is possible for the elements in the same color (or level set) and the

number of elements in the same color (or level set) should be as large as possible in order to obtain high granularity for parallel computation or sufficiently large length of innermost loops for vectorization.

RCM (Fig. 2(a)) reordering provides fast convergence of IC/ILU-preconditioned Krylov iterative solvers, yet with irregular numbers of the elements in each level set. For example in Fig. 2(a), the 1st level set is of size 1, while the 8th level set is of size 8. Multicoloring provides a uniform element number in each color (Fig. 2(b)). However, it is widely known that the convergence of IC/ILU-preconditioned Krylov iterative solvers with multicoloring is rather slow [4]. Convergence can be improved by increasing the number of colors because of fewer incompatible local graphs [4], but this reduces the number of elements in each color. The solution for this trade-off is cyclic multicoloring (CM) on RCM [16]. In this method, the elements are renumbered in a cyclic manner. Figure 2(c) shows an example of CM-RCM reordering. In this case, there are 4 colors; the 1st, 5th, 9th and 13th colors in Fig. 2(a) are classified into the 1st color. There are 16 elements in each color. In CM-RCM, the number of colors should be large enough to ensure that elements in the same color are independent.

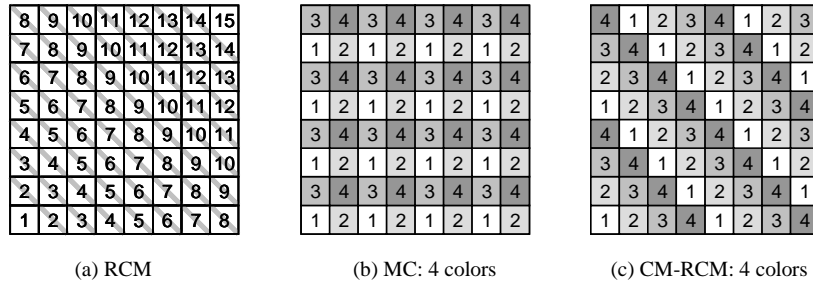


Fig. 2 Example of hyperplane/RCM, multicoloring and CM-RCM reordering for 2D geometry [10]

### 3.2 DJDS Reordering

The compressed row storage (CRS) [1] matrix storage format is highly memory-efficient, however the innermost loop is relatively short, order of number of off-diagonal components, for matrix-vector operations as shown below:

```

do i= 1, N
  do j= 1, NU(i)
    k1= indexID(i, j);k2= itemID(k1)
    F(i)= F(i) + A(k1)*X(k2)
  enddo
enddo

```

The following loop exchange is then effective for obtaining a sufficiently long innermost loop for vector operations [5]:

```

do j= 1, NUmex
  do i= 1, N
    k1= indexID(i, j);k2= itemID(k1)
    F(i)= F(i) + A(k1)*X(k2)
  enddo
enddo

```

Descending-order jagged diagonal storage (DJDS) [16] is suitable for this type of operation and involves permuting rows into an order of decreasing number of non-zeros, as in Fig. 3(a). As elements on the same color are independent, performing this permutation inside each color

does not affect results. Thus, a 1D array of matrix coefficients with continuous memory access and sufficiently long innermost loops can be obtained, as shown in Fig. 3(b).

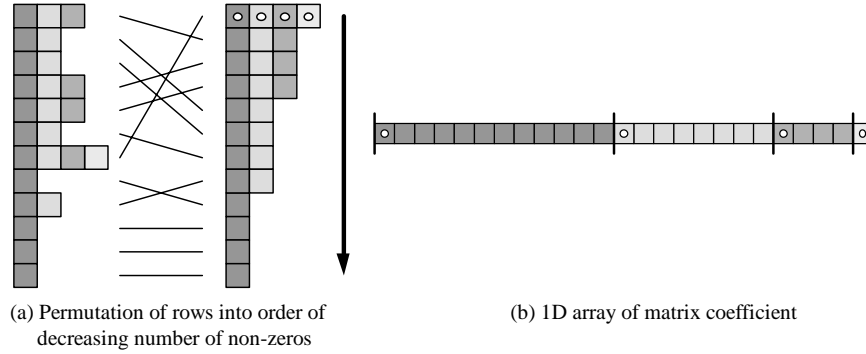


Fig. 3 DJDS reordering for efficient vector/parallel processing

### 3.3 Distribution over SMP Nodes : Parallel DJDS Reordering

The 1D array of matrix coefficients with continuous memory access and sufficiently long innermost loops is suitable for both parallel and vector computing. The loops for this type of array are easily distributed to each PE in an SMP node via loop directives. In order to balance the computational load across PEs in the SMP node, the DJDS array should be reordered again in a cyclic manner. The procedure for this reordering is called parallel DJDS (PDJDS) [10].

### 3.4 Summary of Reordering Methods

The reordering procedures for increasing parallel/vector performance of the SMP cluster architecture described in this section are summarized as follows:

- (1) RCM reordering on the original local matrix for independent sets and CM reordering to obtain innermost loops whose length is sufficiently long and uniform.
- (2) DJDS reordering for efficient vector processing, producing 1D arrays of coefficients with continuous memory access and sufficient length of innermost loops.
- (3) Cyclic reordering for load-balancing among PEs on an SMP node.

The typical loop structure of the matrix-vector operations for PDJDS /CM-RCM or PDJDS/MC reordered matrices based on the vector and OpenMP directives of the Earth Simulator is described as follows. In this study, IC/ILU factorization and matrix-vector product processes are executed in a same manner using same ordering methods and indexes. In the flat MPI programming model, `PEsmptTOT` is set to 1 without any option of OpenMP for compiler while `PEsmptTOT` is set to 8 in the hybrid programming model for the Earth Simulator (Fig.4).

```

do iv= 1, NCOLORS
!$omp parallel do private (iv0,j,is,iE,i,k,kk etc.)
do ip= 1, PEsmptTOT
iv0= STACKmc(PEsmptTOT*(iv-1)+ip- 1)
do j= 1, NLhyp(iv)
is= INL(npLX1*(iv-1)+PEsmptTOT*(j-1)+ip-1)
iE= INL(npLX1*(iv-1)+PEsmptTOT*(j-1)+ip )
!CDIR NODEP
do i= iv0+1, iv0+iE-is
k= i+iS - iv0
kk= IAL(k)
(Important Computations)
enddo
enddo
enddo

```

Fig. 4 Forward/backward substitution procedure for ILU/IC process by PDJDS/CM-RCM reordering.

### 4. Effect of Reordering

The proposed methods were applied to 3D solid mechanics example cases, as described in Fig. 5, which represent linear elastic problems with homogeneous material properties and boundary conditions. Figure 6 shows the results demonstrating the performance on a single SMP node of the Earth Simulator by hybrid parallel programming model. In this case, the following three

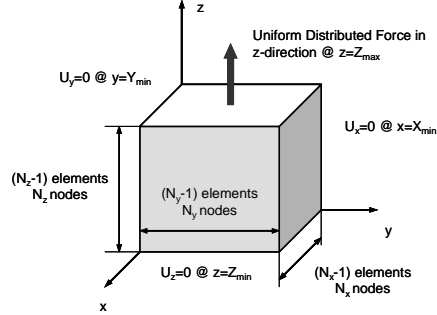


Fig. 5 Problem definition and boundary conditions for 3D solid mechanics example cases.

cases were compared (Fig.6):

- PDJDS/CM-RCM reordering
- Parallel descending-order compressed row storage (PDCRS)/CM-RCM reordering
- CRS without reordering

PDCRS/CM-RCM reordering is identical to PDJDS/CM-RCM except that the matrices are stored in a CRS manner [1] after permutation of rows into the order of decreasing number of non-zeros. The length of the innermost loop is shorter than that for PDJDS. The elapsed execution time was measured for various problem sizes from  $3 \times 10^3$  (12,288) DOF to  $3 \times 10^8$  (6,291,456) DOF on a single SMP node of the Earth Simulator (8 PEs, 64 GFLOPS peak performance, 16 GB memory). The difference between PDCRS and PDJDS for smaller problems is not significant, but PDJDS outperforms PDCRS for larger problems due to larger length of inner-most loops. On the Earth Simulator, the PDCRS performs at a steady 1.5 GFLOPS (2.3% of peak performance), while the performance of PDJDS increases from 3.81 GFLOPS to 22.7 GFLOPS with problem size. The loop length provided by PDCRS is order of number of off-diagonal components for each node, which is less than 30 in this case. On the contrast, average loop length provided by PDJDS is more than 2,500 for the case with  $3 \times 10^8$  (6,291,456) DOF.

The cases without reordering exhibit very poor performance of only 0.30 GFLOPS (0.47% of peak performance). Without reordering, either of parallel and vector computations on a SMP node are impossible for the IC(0) factorization process even in the simple geometry examined in this study. This factorization process represents about 50% of the total computation time in CG solvers with IC(0) preconditioning, as was mentioned before. If this process is not parallelized, the performance decreases significantly.

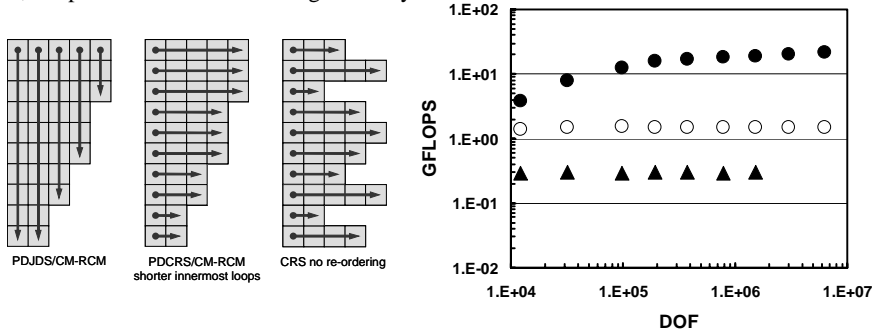
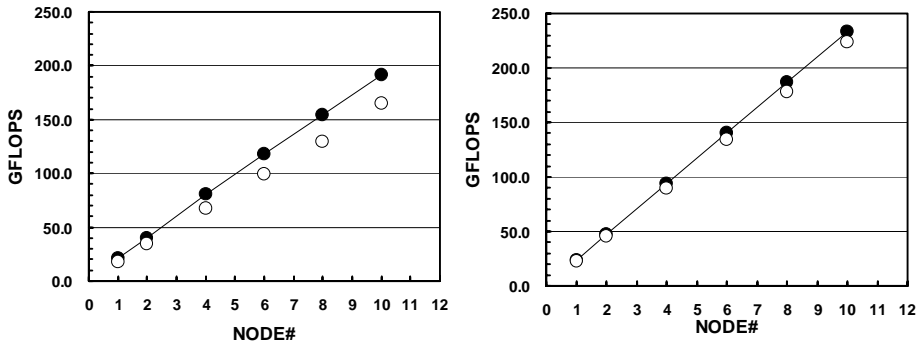


Fig. 6 Effect of coefficient matrix storage method and reordering for the 3D linear elastic problem in Fig.5 with various problem sizes on the Earth Simulator with a single SMP node. (BLACK Circles: PDJDS/CM-RCM, WHITE Circles: PDCRS/CM-RCM, BLACK Triangles: CRS no reordering).

## 5. Performance Evaluation for Large-Scale Problems

Figures 7-10 show the results for large-scale problems having simple geometries and boundary conditions as in Fig. 5 implemented up to 176 SMP nodes of the Earth Simulator (1,408 PEs, 11.26 TFLOPS peak performance, 2.8 TB memory). Performance of the hybrid and flat MPI models were evaluated. The problem size for one SMP node was fixed and the number of nodes was varied between 1 and 176. The largest problem size was  $176 \times 3 \times 128 \times 128 \times 256$  (2,214,592,512) DOF, for which the performance was about 3.80 TFLOPS, corresponding to 33.7 % of the total peak performance of the 176 SMP nodes. The parallel work ratio among SMP nodes for MPI is more than 90% if the problem is sufficiently large.

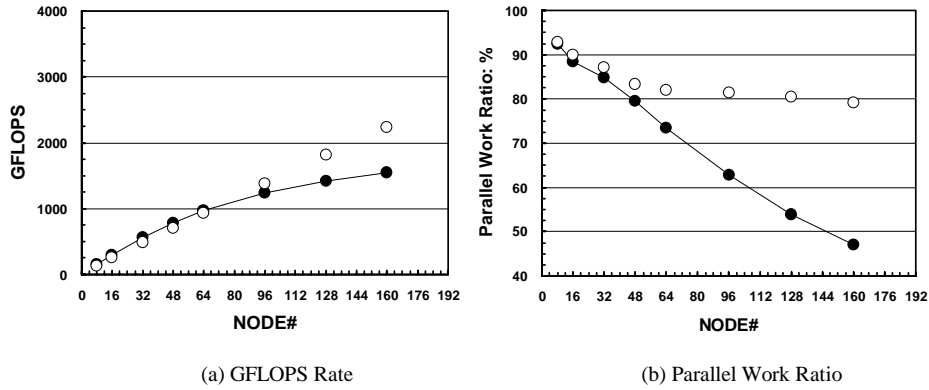
The performance of the hybrid model is competitive with that of the flat MPI model, and both provide robust convergence and good parallel performance for a wide range of problem sizes and SMP node numbers. Iterations for convergence in the hybrid and flat MPI are almost equal, although the hybrid converges slightly faster as shown in Fig. 10(a). In general, flat MPI performs better than the hybrid model for smaller numbers of SMP nodes as shown in Fig. 7, while the hybrid outperforms flat MPI when a large number of SMP nodes are involved (Fig.8-10), especially if the problem size per node is small as shown in Fig.8 and Fig.10. This is mainly because of the latency overhead for MPI communication. According to the performance estimation for finite-volume application code for CFD with local refinement in [8], a greater percentage of time is required by the latency component on larger processor counts, simply due to the available bandwidth being much larger (Fig. 11). Flat MPI requires eight times as many MPI processes as hybrid model. If the node number is large and problem size is small, this effect is significant.



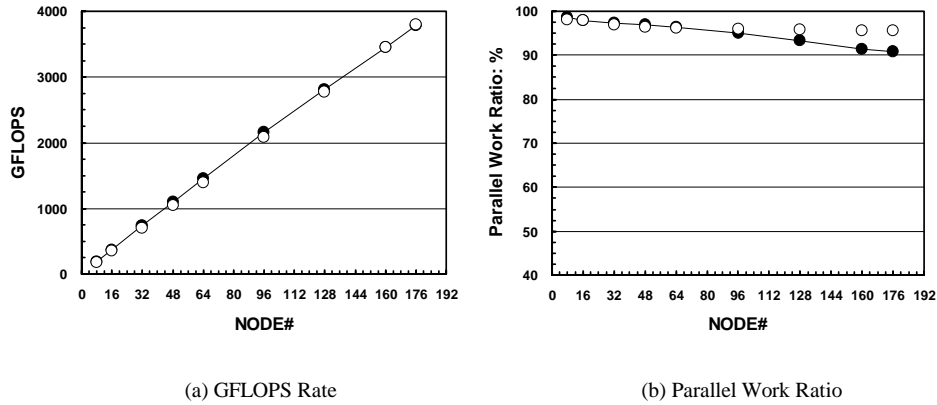
(a) Problem size/SMP = 786,432 DOF ( $3 \times 64^3$ ). Largest case is 7,864,320 DOF on 10 SMP nodes (80 PEs). Maximum performance is 192 (Flat MPI) and 165 (Hybrid) GFLOPS.

(b) Problem size/SMP node = 12,582,912 DOF ( $3 \times 256 \times 128 \times 128$ ). Largest case is 125,829,120 DOF on 10 SMP nodes (80 PEs). Maximum performance is 233 (Flat MPI) and 224 (Hybrid) GFLOPS

**Fig. 7** Problem size and parallel performance on the Earth Simulator for the 3D linear elastic problem in Fig.5 using between 1 and 10 SMP nodes. (BLACK Circles: Flat MPI, WHITE Circles: Hybrid). PDJDS/CM-RCM reordering.

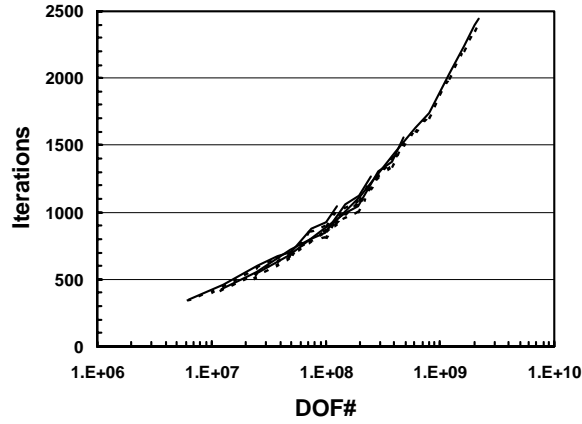


**Fig. 8** Problem size and parallel performance on the Earth Simulator for the 3D linear elastic problem in Fig.5 using between 8 and 160 SMP nodes. (a)GFLOPS rate and (b)Parallel work ratio. Problem size/PE is fixed as 786,432 DOF ( $3 \times 64^3$ ). Largest case is 125,829,120 DOF on 160 SMP nodes (1280 PEs). Maximum performance is 1.55 (Flat MPI) and 2.23 (Hybrid) TFLOPS (Peak performance= 10.24 TFLOPS). (BLACK Circles: Flat MPI, WHITE Circles: Hybrid). PDJDS/CM-RCM reordering.

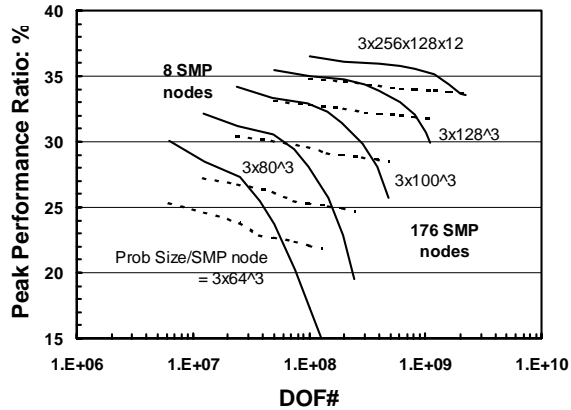


**Fig. 9** Problem size and parallel performance on the Earth Simulator for the 3D linear elastic problem in Fig.5 using between 8 and 176 SMP nodes. (a)GFLOPS rate and (b)Parallel work ratio. Problem size/SMP node is fixed as 12,582,912 DOF ( $3 \times 256 \times 128 \times 128$ ). Largest case is 2,214,592,512 DOF on 176 SMP nodes (1408 PEs). Maximum performance is 3.78 (Flat MPI) and 3.80 (Hybrid) TFLOPS (Peak performance= 11.26 TFLOPS). (BLACK Circles: Flat MPI, WHITE Circles: Hybrid). PDJDS/CM-RCM reordering.

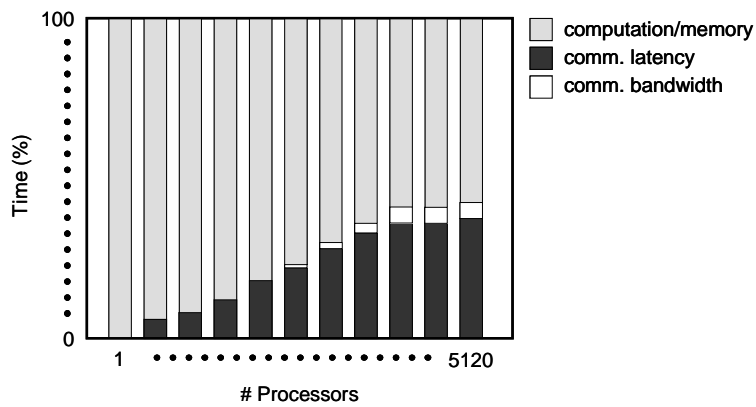
(a) Iterations for Convergence



(b) Peak Performance Ratio



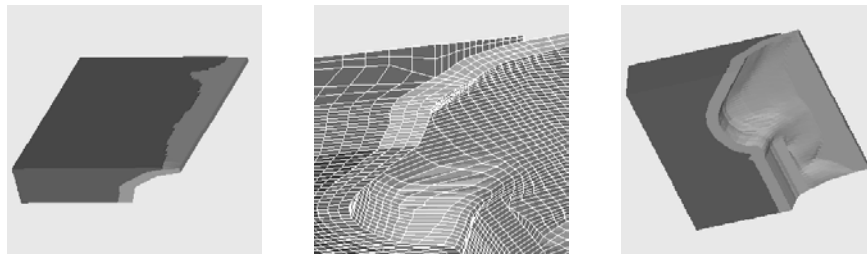
**Fig. 10** Problem size and parallel performance on the Earth Simulator for the 3D linear elastic problem in Fig.5 using between 8 and 176 SMP nodes. (a) Iterations for convergence and (b) Ratio to the peak performance. Problem size/SMP node is fixed (THICK lines: Flat MPI, DASHED lines: Hybrid). PDJDS/CM-RCM reordering.



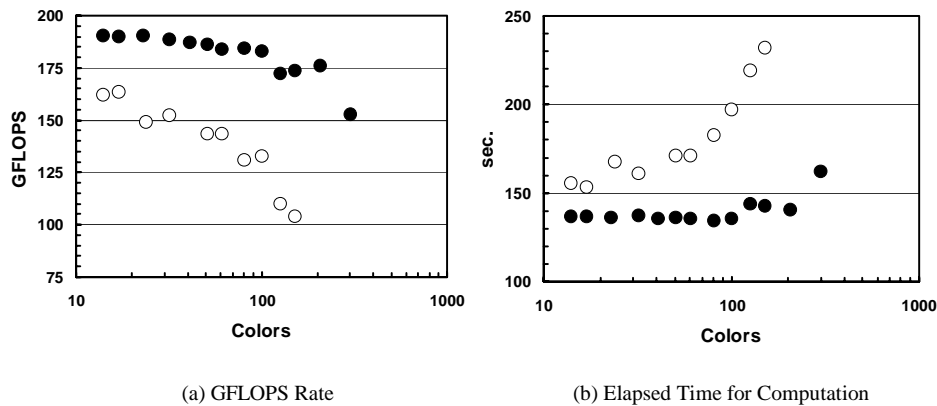
**Fig. 11** Performance estimation of a finite-volume application code for CFD with local refinement on the Earth Simulator. Based on the results described in [8]. A greater percentage of time is taken by the latency component on larger processor counts, simply due to its much larger available bandwidth.

## 6 Solid Earth Simulations for Complicated Geometries and Boundary Conditions

Figures 12 and 13 show the most recent results in solid earth simulations (ground motion simulation with contact) on the Earth Simulator using parallel iterative methods and parallel programming models developed in this study [11]. Figure 12 describes the model of the Southwest Japan, consisting of 7,767,002 nodes (23,301,006 DOF) and 7,684,072 tri-linear (1st order) hexahedral elements. Special preconditioning method, called selective blocking, has been developed based on the block IC/ILU method in this study for non-linear problems due to fault-zone contact with penalty constraint conditions [11]. Figure 13 provides preliminary results of the simulation on the Earth Simulator. Effects of color numbers for multicolor reordering [11] in flat MPI and three-level hybrid are shown. Both programming models attained about 30% of the peak performance (640 GFLOPS) even in this complicated geometry with contact boundary conditions, while flat MPI shows better performance.



**Fig. 12** Description of the Southwest Japan model. This model consists of crust (dark gray) and subduction plate (light gray). 7,767,002 nodes (23,301,006 DOF) and 7,684,072 tri-linear (1st order) hexahedral elements are included [11].



**Fig. 13** Performance on 10 SMP nodes of the Earth Simulator (peak performance = 640 GFLOPS) using CG iterative method with *selective blocking* preconditioning [11] for the 3D elastic contact problem with MPC (linear multiple point constraint) condition ( $\lambda=10^6$ ) in Fig.12 (Southwest Japan model, 23,301,006 DOF). Effect of color numbers for multicolor reordering. (a) GFLOPS rate, (b) Elapsed Time for Computation. (BLACK Circles: Flat MPI, WHITE Circles: Hybrid). PDJDS/multicolor (MC) reordering [11].

## 7 Conclusions and Remarks

This paper described an efficient parallel iterative method for unstructured grids developed for SMP cluster architectures with vector processors, such as the Earth Simulator. The method employs three-level hybrid parallel programming model consisting of the following hierarchy, (1) Inter-SMP node: MPI, (2) Intra-SMP node: OpenMP for parallelization and (3) Individual PE: Compiler directives for vectorization. Multiple reordering methods have been applied in order to attain concurrent local operations, no global dependency, continuous memory access, and sufficiently long innermost loops. PDJDS (parallel descending-order jagged diagonal storage) provides long innermost loops. CM-RCM (cyclic-multicoloring/reverse Cuthill-McKee) provides local and parallel operations.

Simple 3D linear elastic problems with more than  $2.2 \times 10^9$  DOF were solved by  $3 \times 3$  block ICCG(0) with additive Schwarz domain decomposition and PDJDS/CM-RCM reordering on 176 SMP nodes of the Earth Simulator, achieving performance of 3.80 TFLOPS (33.7 % of peak performance). PDJDS/CM-RCM reordering provides excellent vector and parallel performance on SMP nodes. Without reordering, parallel processing of forward/backward substitution in IC/ILU factorization was impossible due to global data dependencies even in the simple examples in this study. While the three-level hybrid and flat MPI parallel programming models offer similar performance, the hybrid programming model outperforms flat MPI in the problems with large numbers of SMP nodes. In the next stage, developed solvers will be applied to realistic applications with complicated geometries and boundary conditions in science and engineering field, as shown in the previous section.

## Acknowledgements

This study is a part of the "*Solid Earth Platform for Large-Scale Computation*" project funded by the Ministry of Education, Culture, Sports, Science and Technology, Japan through Special Promoting Funds of Science & Technology.

## References

1. Barrett, R., Berry, M., Chan, T.F., Donato, J., Dongarra, J.J., Eijkhout, V., Pozo, R., Romine, C. and van der Vorst, H. : Templates for the Solution of Linear Systems : Building Blocks for Iterative Methods, SIAM, 1994.
2. Cappelto, F. and Etiemble, D. : "MPI versus MPI+OpenMP on the IBM SP for the NAS Benchmarks", SC2000 Technical Paper, Dallas, Texas, 2000.
3. Djomehri, M.J. and Jin, H.H.: "Hybrid MPI+OpenMP Programming of an Overset CFD Solver and Performance Investigations", NASA/NAS Technical Report (NASA Ames Research Center), NAS-02-002, (2002). 16.
4. Doi, S. and Washio : "Using Multicolor Ordering with Many Colors to Strike a Better Balance between Parallelism and Convergence", RIKEN Symposium on Linear Algebra and its Applications, The Institute of Physical and Chemical Research, 1999, pp.19-26.
5. Dongara, J.J., Duff, I.S., Sorensen, D.C. and van der Vorst, H.A. : Solving Linear Systems on Vector and Shared Memory Computers, SIAM, 1990.
6. Falgout, R. and Jones, J. : "Multigrid on Massively Parallel Architectures", Sixth European Multigrid Conference, Ghent, Belgium, September 27-30, 1999.
7. Garatani, K., Nakamura, H., Okuda, H., Yagawa, G.: "GeoFEM: High Performance Parallel FEM for Solid Earth", HPCN Europe 1999, Amsterdam, The Netherlands, Lecture Notes in Computer Science, Vol.1593 (1999) pp.133-140.
8. Kerbyson, D.J., Hoisie, A. and Wasserman, H. "A Comparison Between the Earth Simulator and Alpha Server Systems using Predictive Application Performance Models", LA-UR-02-5222, Los Alamos National Laboratory, USA, 2002.
9. Nakajima, K., Okuda, H.: "Parallel Iterative Solvers with Localized ILU Preconditioning for Unstructured Grids on Workstation Clusters", International Journal for Computational Fluid Dynamics, Vol.12 (1999) pp.315-322.
10. Nakajima, K. and Okuda, H.: "Parallel Iterative Solvers for Unstructured Grids using an OpenMP/MPI Hybrid Programming Model for the GeoFEM Platform on SMP Cluster Architectures", International Workshop on OpenMP: Experiences and Implementations (WOMPEI 2002), Lecture Notes in Computer Science 2327 (2002), pp.437-448.
11. Nakajima, K. : "Parallel Iterative Solvers of GeoFEM with Selective Blocking Preconditioning for Nonlinear Contact Problems on the Earth Simulator", RIST/Tokyo GeoFEM Report 2003-006 ([http://geofem.tokyo.rist.or.jp/report\\_en/2003\\_006.html](http://geofem.tokyo.rist.or.jp/report_en/2003_006.html)), 2003.
12. Olikar, L, Li X., Husbands, P. and Biswas, R.: "Effects of Ordering Strategies and Programming Paradigms on Sparse Matrix Computations", SIAM Review, Vol.44, No. 3(2002), pp.373-393.
13. Rabenseifner, R.: "Communication Bandwidth of Parallel Programming Models on Hybrid Architectures", International Workshop on OpenMP: Experiences and Implementations (WOMPEI 2002), Lecture Notes in Computer Science 2327 (2002), pp.437-448.
14. Saad, Y. : Iterative Methods for Sparse Linear Systems, PWS Publishing Company, 1996.
15. Smith, B., Björstam, P. and Gropp, W.: Domain Decomposition, Parallel Multilevel Methods for Elliptic Partial Differential Equations, Cambridge Press, 1996.
16. Washio, T., Maruyama, K., Osoda, T., Shimizu, F. and Doi, S. : "Efficient implementations of block sparse matrix operations on shared memory vector machines", SNA2000 : The Fourth International Conference on Supercomputing in Nuclear Applications, 2000.
17. Accelerated Strategic Computing Initiative (ASCI) Web Site : <http://www.llnl.gov/asci/>
18. Earth Simulator Center Web Site: <http://www.es.jamstec.go.jp/>
19. GeoFEM Web Site: <http://geofem.tokyo.rist.or.jp/>
20. MPI Web Site : <http://www.mpi.org/>
21. OpenMP Web Site : <http://www.openmp.org/>