The Impact of Parallel Programming Models on the Linear Algebra Performance for Finite Element Simulations

Kengo Nakajima[1]

[1] Department of Earth and Planetary Science, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 112-0002, Japan
`nakajima@eps.s.u-tokyo.ac.jp`
`http://www-solid.eps.s.u-tokyo.ac.jp/~nakajima`

Abstract. Parallel iterative linear solvers for unstructured grids in FEM applications, which were developed for the Earth Simulator (ES), have been ported to various types of parallel computers. Performance of flat-MPI and hybrid parallel programming model has been compared for ES, Hitachi SR8000, IBM SP-3 and IBM p5-model 595. Effect of coloring and method for storage of coefficient matrices have been also evaluated in various types of applications. Performance with more than 10^4 processors has been estimated using measured data up to 10^3 processors.

1 Introduction

1.1 Parallel programming models on SMP cluster architectures

Recently, symmetric multiprocessor (SMP) cluster architectures have become very popular as teraflop-scale parallel computers, such as the DOE-ASC (Advanced Simulation & Computing, formerly *ASCI*) [1] machines and the Earth Simulator (ES) [2].

In order to achieve minimal parallelization overhead, a multi-level *hybrid* programming model is often employed for this architectures (Fig.1). In this method, coarse-grain parallelism is achieved through domain decomposition by message passing among SMP nodes, and fine-grain parallelism is obtained by loop-level parallelism inside each SMP node by compiler-based thread parallelization such as OpenMP.

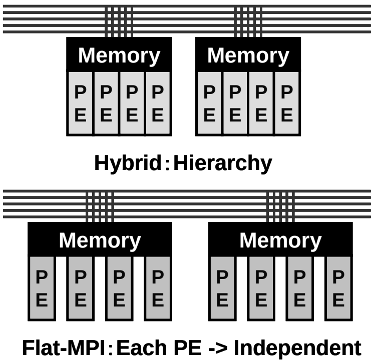
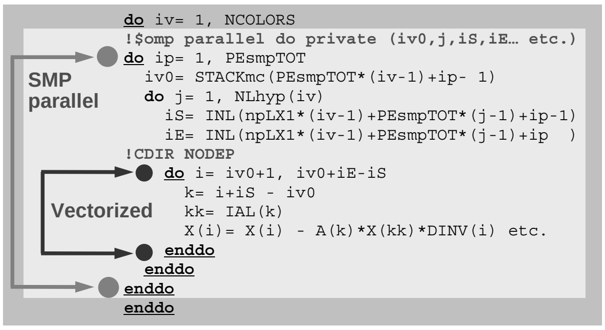


Fig.1 Parallel programming models for SMP cluster architectures

Another often-used programming model is the single-level *flat-MPI* model (Fig.1), in which separate single-threaded MPI processes are executed on each processing element (PE). The efficiency of each model depends on hardware performance (CPU speed, communication bandwidth, memory bandwidth and their balance), features of applications, and problem size [3].

1.2 Previous works

In the previous works [4,5], the author developed parallel iterative linear solvers for unstructured grids of finite-element applications in GeoFEM [6] on the ES using both of the *flat-MPI* and *hybrid* parallel programming models. Multicolor and reverse Cuthill-Mckee (RCM) ordering techniques [7,8] provide excellent parallel and vector performance on the ES for iterative solvers with ILU/IC-type preconditioning. The flat-MPI and hybrid parallel programming models are competitive in most cases. Hybrid outperforms Flat-MPI when number of SMP node is large and problem size is not so large. This is estimated because of the effect of communication latency in many MPI processes [9]. In the cases with many colors, fewer numbers of iterations are required for convergence, but the performance is rather worse due to the smaller loop length and greater overhead. The hybrid parallel programming model is much more sensitive to color number than the flat-MPI. If the color number increases in the hybrid programming model, the frequency of *do-loops* with OpenMP for SMP unit increases, as shown in Fig.2 [4,5]. Finally, synchronization overhead of OpenMP is significant [4,5].



```
do iv= 1, NCOLORS
!$omp parallel do private (iv0,j,iS,iE… etc.)
do ip= 1, PEsmpTOT
  iv0= STACKmc(PEsmpTOT*(iv-1)+ip- 1)
  do j= 1, NLhyp(iv)
    iS= INL(npLX1*(iv-1)+PEsmpTOT*(j-1)+ip-1)
    iE= INL(npLX1*(iv-1)+PEsmpTOT*(j-1)+ip  )
!CDIR NODEP
    do i= iv0+1, iv0+iE-iS
      k= i+iS - iv0
      kk= IAL(k)
      X(i)= X(i) - A(k)*X(kk)*DINV(i) etc.
    enddo
  enddo
enddo
enddo
```

Fig.2 Forward/backward substitution procedure using OpenMP and vectorization directives during ILU(0)/IC(0) preconditioning [4,5]

In GeoFEM, localized ILU(0)/IC(0) preconditioning method, which is based on the idea of block Jacobi preconditioner [7], has been mainly applied for parallel iterative solvers [4,5]. Usually, larger number of processors (domains) provide worse convergence rate due to data locality. *Hybrid* parallel programming model was expected to suppress the increase of iteration number for convergence in large scale problems, because the domain number of *hybrid* is 1/8 of that of *flat-MPI* on ES. Results in [4,5] show that this effect is not significant for a wide range of problem size, mainly because of stabilization by additive Schwartz domain decomposition [4,5,6].

1.3 Present work

In the present work, parallel iterative solvers for unstructured grids, developed in [4,5] have been implemented to other supercomputers, Hitachi SR8000/MPP (University of Tokyo) [10], IBM SP-3 (NERSC/LBNL) [11] and IBM p5-model 595 (Kyushu University) [12]. Effect of color number and method for storage of coefficient matrices on performance have been evaluated through benchmarks based on real applications, using single processing element (PE), single SMP node, and multiple nodes, for both of flat-MPI and hybrid parallel programming models. Performance with more than 10^4 processors has been estimated using measured data up to 1,000 PE's. Recently, various SMP clusters have been evaluated using applications with unstructured grids, such as finite-element method (FEM) [13,14]. But, they are mainly focused on the flat-MPI programming model.

2 Overview of hardware and software environments

2.1 Hardware

Table 1 presents a summary of the architectural characteristics of the three supercomputers in the current work.

Table 1 Architectural highlights of Earth Simulator, Hitachi SR8000, and IBM SP-3 platforms

	Earth Simulator[2]	Hitachi SR8000 [10]	IBM SP-3 [11]	IBM p5-595 [12]
PE#/node	8	8	16	16
Clock rate (MHz)	500	450	375	1,900
Peak performance/PE (GFLOPS)	8.00	1.80	1.50	7.60
Memory/node (GB)	16	16	16~64	64~128
Memory BW (GB/sec)	32	4	1	6.4
Network BW (GB/sec/node)	12.3	1.60	1.00	4.00
MPI Latency (μsec)	5.6-7.7 [15]	6-20 [16]	16.3 [14]	3.9 [12]

Earth simulator [2] is a parallel-vector systems based on NEC SX-6, with 640 SMP nodes, 5,120 vector processors, and 10 TB memory. Total peak performance is 40 TFLOPS. Each node is connected through single-stage crossbar network.

Hitachi SR8000/MPP (Hitachi SR8000) at University of Tokyo, based on Hitachi SR8000 model G1[10], has very similar architecture with that of ES. Entire system has 128 SMP nodes, 1,024 Power3-based processors and 2 TB memory. Total peak performance is 1.84 TFLOPS. Each PE is a scalar processor, but provides excellent performance on codes for vector processors through its pseudo-vector capability [10]. Each SMP node is connected through three-dimensional crossbar network.

The IBM SP-3 at NERSC/LBNL (Seaborg) [11] is a POWER3-based super scalar system, with 380 SMP nodes, 6,080 processors, and 7.3 TB memory. Total peak performance is 9.12 TFLOPS. Each PE has a 64 KB Level-1 data cache and 8 MB Level-2 cache. Multi-node configurations are networked via the Colony switch. In this study, only 8 PE's of 16 PE's on each SMP node have been used for comparison with ES and Hitachi SR8000.

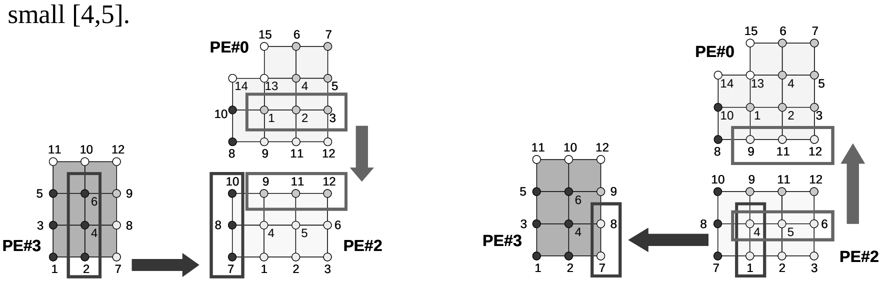
The IBM p5-model 595 (IBM p5-595) at Kyushu University [12] is a POWER5-based super scalar system, with 26 SMP nodes, 416 processors, and 2.0 TB memory. Total peak performance is 3.16 TFLOPS. In this study, only one SMP node has been used, and 8 PE's of 16 PE's on a SMP node have been used. Each PE has an 18 MB Level-3 cache.

2.2 Software

In the present work, parallel iterative solvers with preconditioning for various types of applications on unstructured grids, which were developed for ES, have been evaluated. The following two types of preconditioners have been considered [5,6]:

I. Localized block ILU(0) preconditioning method for 3D solid mechanics.
II. Selective blocking preconditioning method for 3D solid mechanics with contact conditions [5,17].
III. Parallel multigrid preconditioning method for 3D Poisson equations derived from incompressible Navier-Stokes solvers with adaptive meshes [5]

Local data structure in GeoFEM has been applied [6]. In FEM-type applications, most of communications among processors occur at exchanging information on domain boundaries (Fig.3). Ratio of communication compared to computation is usually small [4,5].



(a) SEND (b) RECEIVE

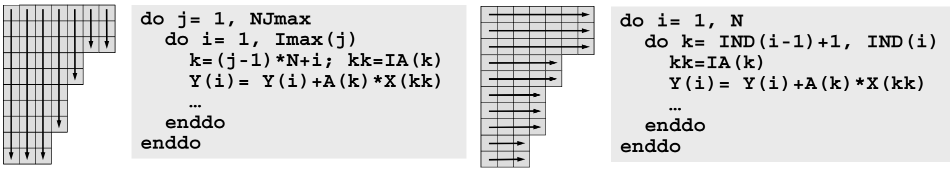
Fig.3 Communication among processors in parallel FEM [4,5]

The codes are optimized for vector processors using multicolor-based reordering techniques [4,5,8], which provide, (1) local operations and no global dependency, (2) contiguous memory access, and (3) sufficiently long loops.

In the hybrid parallel programming model, the following three levels of parallelism are considered, (1) MPI for inter-SMP node communication, (2) OpenMP for intra node prallelization, and (3) compiler directives for vectorization of individual PE.

Coefficient matrices have been stored according to descending order jagged diagonal manner (DJDS) (Fig.4(a)) in the original code [4,5]. This method provides long innermost loops, and is suitable for vector processors. In this work, descending order compressed row storage (DCRS) (Fig.4(b)) is also tested for Hitachi SR8000, IBM SP-3, and IBM p5-595. DCRS provides rather shorter loops than DJDS, but reduction-type innermost loops of DCRS attain good data locality, which is advantageous for cache utilization [18]. DJDS and DCRS require same iteration number for convergence as long as same color number has been applied.

The procedure for forward/backward substitution (FBS) using OpenMP and vectorization directives during ILU(0)/IC(0) preconditioning by DJDS/MC ordering (DJDS with multicoloring) is shown in Fig.2. In the flat-MPI programming model, **`PEsmpTOT`** is set to 1 without any options of OpenMP for compiler while **`PEsmpTOT`** is set to 8 in the hybrid programming model for ES.



```
do j= 1, NJmax
  do i= 1, Imax(j)
    k=(j-1)*N+i; kk=IA(k)
    Y(i)= Y(i)+A(k)*X(kk)
    …
  enddo
enddo
```

```
do i= 1, N
  do k= IND(i-1)+1, IND(i)
    kk=IA(k)
    Y(i)= Y(i)+A(k)*X(kk)
    …
  enddo
enddo
```

(a) DJDS (b) DCRS

Fig.5 Storage scheme and loop organization for matrix operations

3. Single PE/SMP node performance

Table 2 shows single PE performance of ICCG solver using DJDS with multicolor ordering for simple FEM application in 3D linear elastic solid mechanics for simple cubic geometry [4,5] with homogeneous isotropic material properties and boundary conditions.

Hybrid parallel programming model with a single thread has been applied. Estimated performance has been derived according to peak performance and memory bandwidth by the method described in [19]. Effect of cache in the IBM SP-3 and IBM p5-595 is not considered here. Number of color is 8, and problem size is 786,432 DOF (3×64^3). In each supercomputer, measured performance agrees with estimated one very well.

Figure 5 shows the results demonstrating the performance on a single SMP node (8 PE's). The elapsed execution time was measured for various problem sizes from 3×16^3 (12,288) DOF to 3×128^3 (6,291,456) DOF. Color number has been fixed as 100. On the ES and Hitachi SR8000, DJDS outperforms DCRS for larger problems due to larger length of innermost loops. On ES, the performance of DJDS increases from 3.81 GFLOPS to 22.7 GFLOPS (from 6.0% to 35.5% of the peak performance) with problem size. Pseudo-vector capability of Hitachi compiler provides good performance. On the IBM SP-3 and IBM p5-595, difference between DJDS and DCRS is not significant, and performance is better for small problem size due to the effect of cache. DCRS is better than DJDS for small problem size, because DCSR utilizes cache more effectively.

On the ES, the flat-MPI and hybrid are competitive, but flat-MPI is slightly better for DJDS. On Hitachi SR8000, the hybrid is much better. On the IBM SP-3 and IBM p5-595, they are competitive if problem size is large, but the flat-MPI is much better for small problems, especially for the IBM SP-3. Cache on each processor is utilized more efficiently in flat-MPI parallel programming model. Reasons for difference between the flat-MPI and hybrid on the Hitachi SR8000 with DJDS are not clear. According to the investigation in [20], pseudo-vector capability does not seem to work efficiently in the flat-MPI for large problem size.

Table 2 Single CPU performance for finite-element type applications of each architecture

	Earth Simulator	Hitachi SR8000	IBM SP-3	IBM p5-595
Peak performance/PE (GFLOPS)	8.00	1.80	1.50	7.60
Memory BW (GB/sec)	32	4	1	6.4
Estimated performance (GFLOPS (% of peak))	2.62-3.89 (32.7-48.6)	.383-.486 (21.3-27.0)	.113-.122 (7.50-8.11)	.314-.775 (4.14-10.2)
Measured performance	3.33 (41.6)	.432 (24.0)	.128 (8.53)	.501 (6.59)

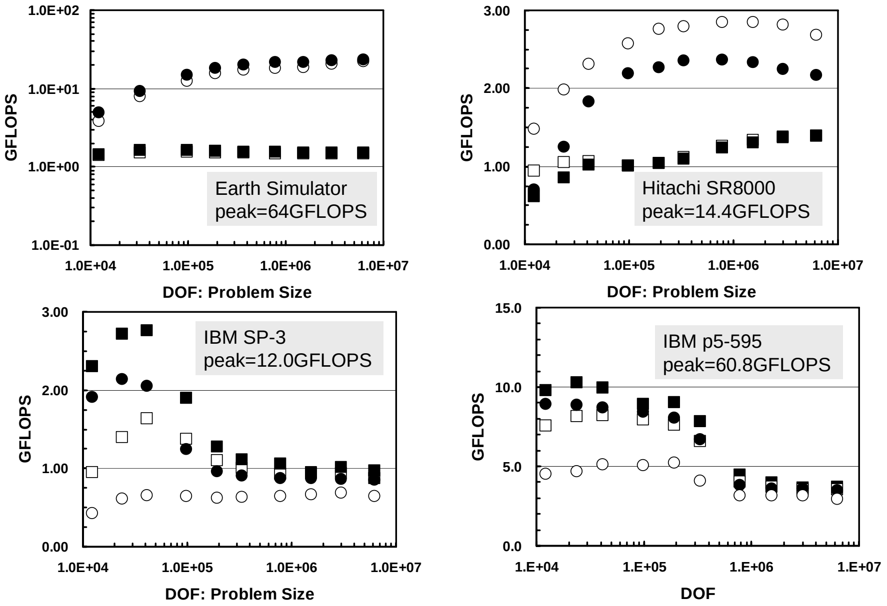


Fig. 5 Effect of coefficient matrix storage method and the flat-MPI/hybrid for the 3D linear elastic problem for simple cubic geometry with various problem sizes on a single SMP node (8 PE's) (100 colors)

4 Effect of color number

Convergence of iterative solvers using multicolor reordering method can be improved by increase of color number, because of fewer incompatible local graphs [8]. But this reduces the number of elements in each color, which means shorter innermost loops for vectorization [4,5,8]. In this section, this effect is investigated for both of the flat-MPI and hybrid programming models using a single SMP node (8 PE's) of each supercomputer for various types of applications and geometries.

4.2 Elastic solid mechanics

First example is the simple 3D linear elastic problem for cube in [4,5] with 3×10^6 DOF (3×100^3). Figure 6 shows the effect of color number on convergence of ICCG solvers using DJDS and DCRS with multicolor ordering. Iteration number for convergence decreases as color number increases in both of the flat-MPI and hybrid. The hybrid programming model requires slightly fewer numbers of iterations for convergence.

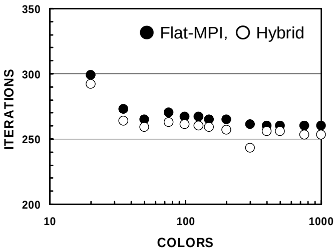


Fig.6 Effect of color number in multicolor reordering: Iteration number for convergence on 3D linear elastic problem for cube (3×10^6 DOF (3×100^3) using 1 SMP node (8 PE's)

Figure 7 shows effect of color number on performance of the ES for DJDS. In both of the flat MPI and hybrid, GFLOPS rate decreases as color number increases. Therefore, elapsed time for computation is longer for 1,000 color cases, although iteration number decreases, as shown in Fig.6. This feature is much more significant in the *hybrid*, as shown in Fig.7. Size of vector register in the ES is 256 [2]. In this case with 10^6 finite-element nodes on 8 PE's, average innermost loop-length is 256 for the case with 488 colors. But, Fig.7 shows that the performance of the hybrid decreases when the color number is about 100. This is mainly because of synchronization overhead of OpenMP in FBS loop of ILU/IC factorization (Fig.4) [4,5].

On the Hitachi SR8000, performance of the hybrid programming model slightly decreases in many color cases due to synchronization overhead of OpenMP, as shown in Fig.8. But, this feature is not so significant as on the ES. In flat-MPI with DJDS, performance becomes larger, as color number increases. On the IBM SP-3 and IBM p5-595, effect of color number on performance is not clear (Fig.8). DCRS is slightly better, and DJDS seems more sensitive to color number than DCRS. Performance of flat-MPI with DJDS is improved according to increase of color number. Performance of hybrid with DJDS also increases, as color number increases from 10 to 100, but finally the performance is going worse, as color number increases from 100 to 1,000 due to OpenMP overhead. On IBM p5-595, this drop is not so significant.

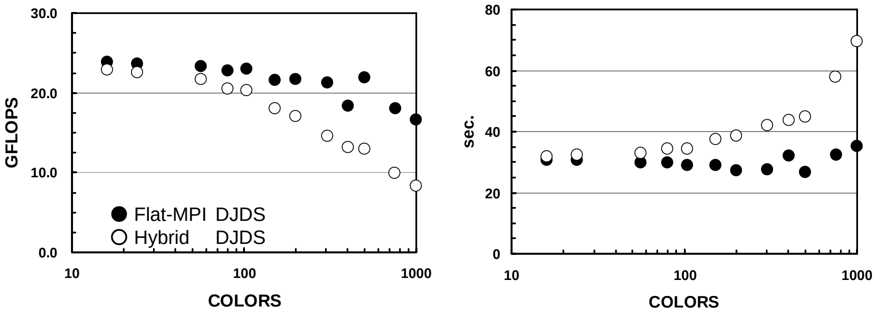
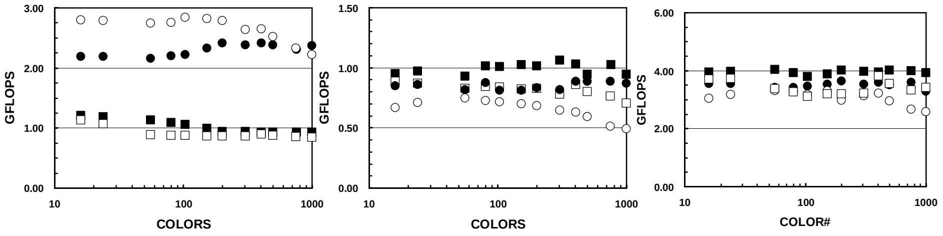


Fig 7 Effect of color number in DJDS ordering on ES with 1 SMP node (8 PE's) for 3D linear elastic problem for cube (Problem size= 3×10^6 DOF (3×100^3), peak=64GFLOPS



(a) Hitachi SR8000 (b) IBM SP-3 (c) IBM p5-595

Fig 8 Effect of color number in DJDS and DCRS ordering on Hitachi SR8000, IBM-SP3 and IBM p5-595 with 1 SMP node (8 PE's) for 3D linear elastic problem for cube (Problem size= 3×10^6 DOF (3×100^3)

4.3 Selective-blocking preconditioning for contact problems

Selective-blocking is a special preconditioning method for contact problems with penalty constraint developed by author. Target application is a simulation of stress accumulation process at plate boundaries around Japan Islands (Fig.9) [4,5,6,17]. In the *selective blocking* method, finite-element nodes in the same contact group coupled through penalty constraints are placed into the large block (*selective block* or *super* node). For symmetric positive definite matrices, block incomplete Cholesky factorization without inter-block fill-in, using *selective blocking* (SB-BIC(0)) shows excellent performance and robustness for a wide range of penalty parameter values [4,5,17].

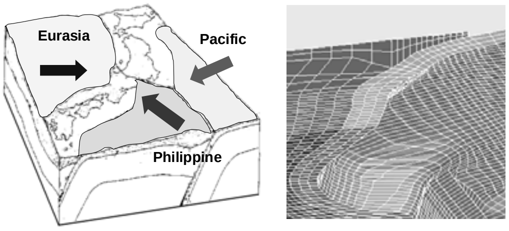


Fig.9 Description of the Southwest Japan model with crust (dark gray) and subduction plate (light gray) [4,5,6,17]

Figure 10 shows the results for the South West Japan model with 784,000 tri-linear hexahedral elements, 823,813 nodes, and 2,471,439 DOF on a single SMP node (8 PE's). Only DJDS ordering has been evaluated. In this geometry, features of relationship between color number and performance, which was mentioned in the previous section, appear more significantly.

On the IBM SP-3 and IBM p5-595, performance becomes better, as color number increases, especially for the flat-MPI. In DJDS ordering, data locality increases as color number increases, because innermost loop in Fig.3(a) becomes shorter in cases with many colors. This means that cache is well utilized in DJDS with many colors on scalar processors. Performance of the hybrid with DJDS also increases, as color number increases from 10 to 100, but finally the performance is going worse, as color number increases from 100 to 1,000 due to OpenMP overhead. On the IBM p5-595, this drop is not so significant.

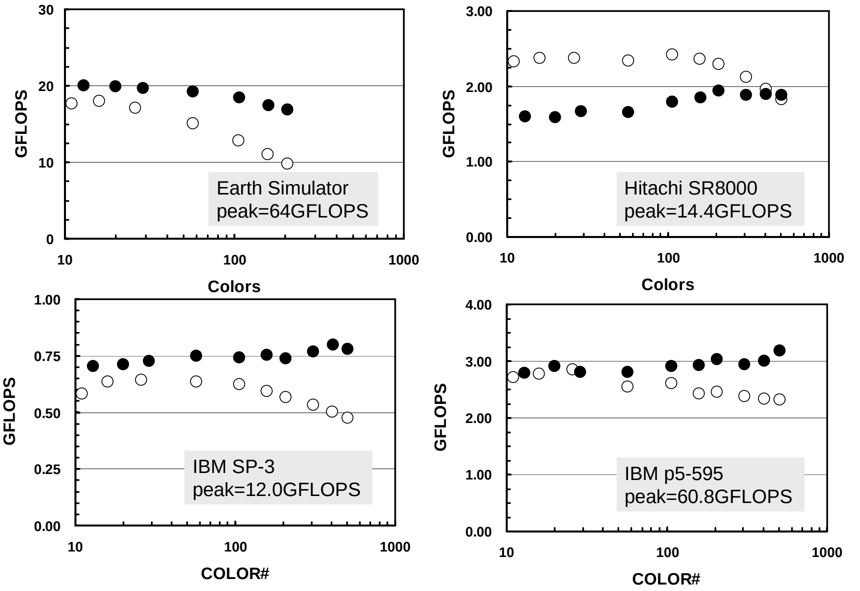


Fig 10 Effect of color number in DJDS ordering on a single SMP node (8 PE's) for 3D contact problem in Fig.9

4.4 Multigrid preconditioning for Poisson equations

Next example is a multigrid-preconditioned conjugate gradient iterative method (MGCG) for Poisson equations described in [5]. Target application is 3D incompressible thermal convection in the region between dual sphere surfaces. This type of geometry appears often in the simulations of earth sciences for both fluid earth (atmosphere and ocean) and solid earth (mantle and outer core). Semi-unstructured prismatic grids generated from triangles on sphere surface are used. Meshes start from icosahedron and are globally refined recursively as in [5]. The grid hierarchy due to recursive refinement can be utilized for the coarse grid formation.

According to the previous works by the author [5], performance drop in the *hybrid* parallel programming model on the ES for many colors was very significant, because of smaller loop length and greater overhead. In this study, same problem has been applied to other hardware. Figure 11 shows performance of MGCG cycles in the Poisson equation with 6,144,000 DOF on a single SMP node (8 PE's). Only DJDS ordering has been evaluated.

In the cases with many colors, fewer numbers of iterations are required for convergence, but the performance is worse due to the smaller loop length and greater overhead. Performance of the ES is much affected by loop length. Moreover, the hybrid parallel programming model is much more sensitive to color number and innermost vector length than the flat-MPI. Results of the Hitachi SR8000, IBM SP-3 and IBM p5-595 provide similar features. On the IBM p5-595, performance drop of hybrid programming model with many colors is not so significant.

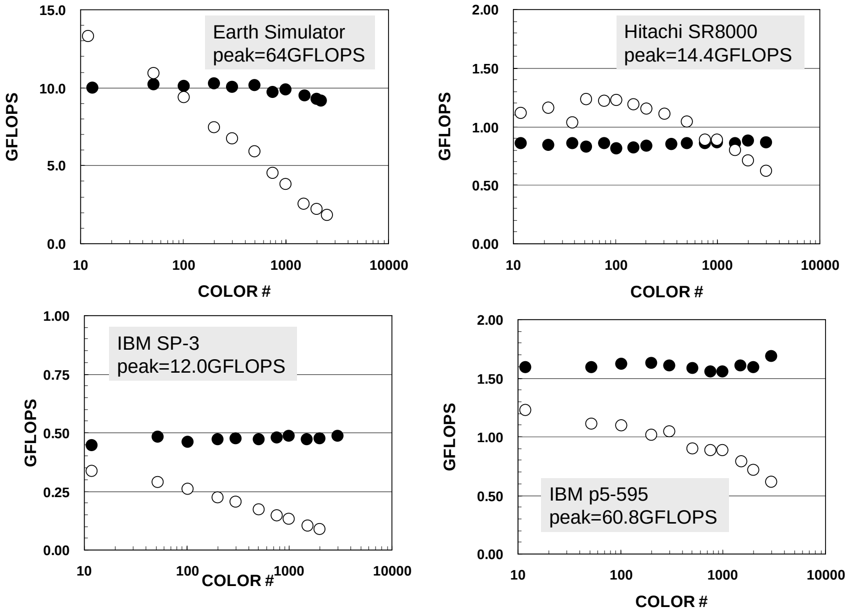


Fig.11 Performance of Poisson solvers with MGCG on a single SMP node (8 PE's) with 6,144,000 cells. (BLACK: Flat-MPI, WHTE: Hybrid)

5 Multiple nodes

Finally, large-scale 3D simple elastic application with simple cubic geometry [4,5] has been solved, using more than 100 SMP nodes of the ES, Hitachi SR8000 and IBM SP-3. Performance of the flat-MPI and hybrid was evaluated. The problem size for one SMP node was fixed and the number of nodes was varied between 1 and 176

(1,408 PE's) for the ES, and between 1 and 128 (1,024 PE's) for the Hitachi SR8000 and IBM SP-3.

On the ES, the largest problem size was 2.21×10^9 DOF, for which the performance was about 3.80 TFLOPS, corresponding to 33.7 % of the total peak performance of the 176 SMP nodes (10.24 TFLOPS) with DJDS (Fig.12) [4,5]. The hybrid and flat-MPI programming models are competitive, but the hybrid outperforms the flat-MPI when a large number of SMP nodes are involved, especially if the problem size per node is small, as shown in Fig.12. Figures 13 and 14 show results obtained by the Hitachi SR8000 with DJDS and the IBM SP-3 with DCRS. On the Hitachi SR8000, the largest problem size was 8.05×10^8 DOF, for which the performance was about 335 GFLOPS, corresponding to 18.2 % of the total peak performance of the 128 SMP nodes. On the IBM SP-3, the largest problem size was 3.84×10^8 DOF, for which the performance was about 110 GFLOPS, corresponding to 7.16 % of the total peak performance of the 128 SMP nodes. In both cases, decrease of performance of the flat-MPI, which happens on ES, has not been observed.

In the current applications, sustained GFLOPS rate for a single SMP node of the ES is 20-30 times as large as that of the IBM SP-3, as shown in Table.2 and Fig.5. Network bandwidth is also 10 times faster. But the rate of MPI latency is very similar. According to [20], if there are 32^3 FEM nodes are on a PE (=98,304 unknowns/PE), computation time for one matrix-vector multiplication procedure (*mat-vec*) for 3D solid mechanics is about 6 msec on the ES if performance is 2.80 GFLOPS/PE (35% of peak). MPI latency of the ES is 6-8 μsec, as shown in Table 1, therefore effect of MPI latency could be very significant in cases with more than 1,000 PE's on the ES.

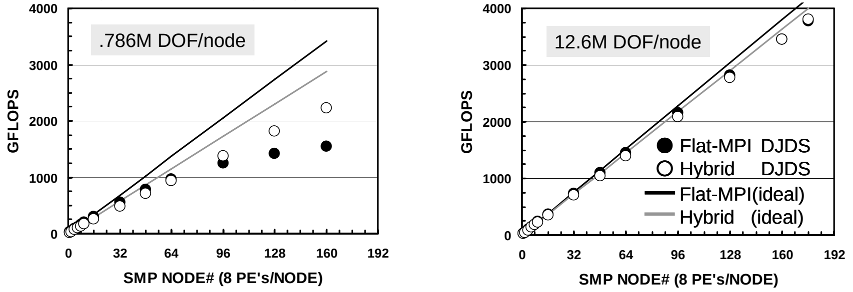


Fig.12 Parallel performance on the ES for the 3D linear elastic problem using between 1 and 176 SMP nodes (1,408 PE's) with DJDS/MC ordering

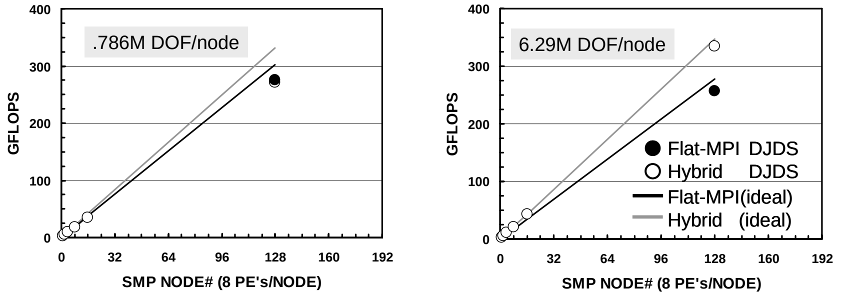


Fig.13 Parallel performance on the Hitachi SR8000 for the 3D linear elastic problem using between 1 and 128 SMP nodes (1,024 PE's) with DJDS/MC ordering

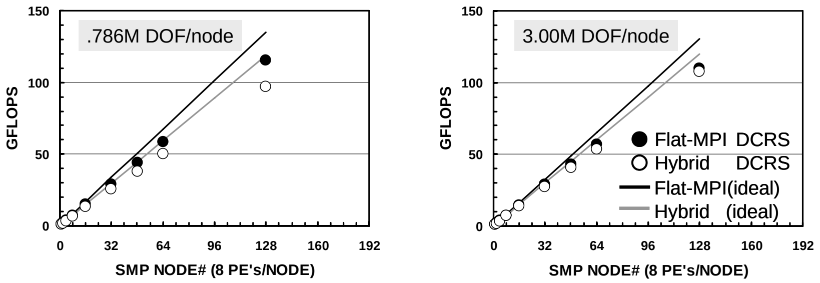


Fig.14 Parallel performance on the IBM SP-3 for the 3D linear elastic problem using between 1 and 128 SMP nodes (1,024 PE's). DCRS/MC ordering

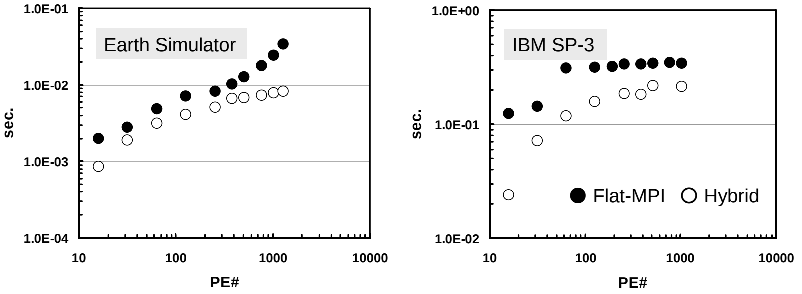


Fig.15 Measured and estimated communication overhead per iteration of the ES and IBM SP-3 with 3×10^6 DOF/SMP-node.

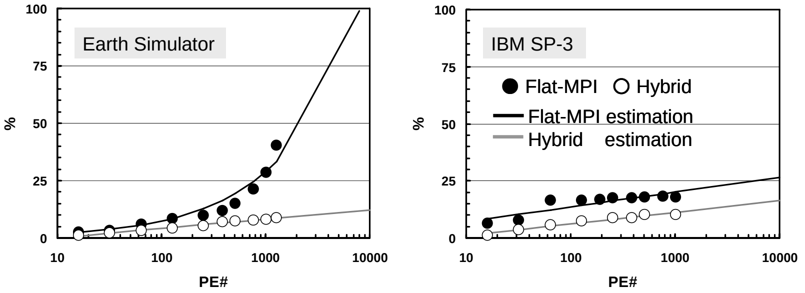


Fig.16 Relative communication overhead of the ES and IBM SP-3 with 3×10^6 DOF/SMP-node. Estimation is based on the extrapolation of measured results. Ratio is based on the elapsed computation time with a single SMP node (8 PE's).

Figures 15 shows measured and estimated communication overhead of the ES and IBM SP-3 with 3×10^6 DOF/SMP-node. Difference between the elapsed computation time per iteration for each case and the result with a single SMP node (8 PE's) is considered as the communication overhead per iteration. Generally, the communication overhead is smaller for the hybrid programming model. On the ES, the communication overhead of the flat-MPI increases constantly, while overhead saturates in other cases for many PE's. Ratio of relative communication overhead to the elapsed computation time with a single SMP node (8 PE's) has been estimated for cases with more than 1,000 PE's according to measured data, as shown in Fig.16. Estimated regression curves are displayed in Fig.16. Measured ratio of overhead with 1,024

PE’s is 28.4 % (ES/flat-MPI), 8.10 % (ES/hybrid), 17.9 % (IBM SP-3/flat-MPI), and 10.1 % (IBM SP-3/hybrid), respectively. Estimated ratio with 10^4 PE’s is 112.8 % (ES/flat-MPI), 12.3 % (ES/hybrid), 26.4 % (IBM SP-3/flat-MPI), and 21.4 % (IBM SP-3/hybrid), respectively. Tests with multiple nodes on the IBM p5-595 have not been conducted, but these numbers will be similar to those of the IBM SP-3 according to balance of performance parameters in Table 1 and single PE/node performance in Table 2 and Fig.5.

6 Concluding Remarks

Parallel iterative linear solvers for unstructured grids in FEM applications developed for the ES have been ported to other SMP cluster supercomputers, such as the Hitachi SR8000, the IBM SP-3 and the IBM p5-595. Performance of the flat-MPI and hybrid parallel programming model has been compared using more than 100 SMP nodes. Effect of coloring and method for storage of coefficient matrices have been also evaluated in various types of applications.

Feature of performance of the Hitachi SR8000 is very similar to that of the ES, because of its pseudo-vector capability, especially in hybrid parallel programming model. Decrease of performance according to color number is not so significant as the ES. On the IBM SP-3, performance is better for small problems. Combination of DCRS and the flat-MPI provides the best performance, because this utilizes cache most efficiently. In DJDS with the flat-MPI, increase of color number provides better performance due to data locality. Features of performance of the IBM p5-595 are similar with those of the IBM SP-3, but performance of the hybrid parallel programming model with OpenMP has been much improved in the IBM p5-595.

The flat-MPI and hybrid parallel programming models are competitive in most cases for each supercomputer. On the ES, the hybrid outperforms the flat-MPI when number of SMP node is large and problem size is small. For example, ratio of communication overhead of ES with 3×10^6 DOF/SMP-node is approximately 40%, when number of PE is 1,280 (160 SMP nodes). This phenomenon has not been observed on the Hitachi SR8000 and the IBM SP-3. This is because of relatively large MPI latency of ES. Parallel performance with more than 1,000 PE’s has been estimated using measured data in this study. According to the estimation, ratio of overhead reaches more than 100 % at 10^4 PE’s on the Earth Simulator with the flat-MPI programming model. Performance of parallel FEM on massively parallel computers strongly depends on the balance of single PE performance, communication latency, and communication bandwidth. Generally, communication overhead with many PE’s is larger in the flat-MPI than in the hybrid programming models.

Acknowledgements

This work is supported by the 21st Century Earth Science COE (Center of Excellence) Program at the University of Tokyo, and CREST/Japan Science and Technology Agency (JST). The author would like to thank the Earth Simulator Center, Information Technology Center at the University of Tokyo, National Energy Research Scientific Computing Center at Lawrence Berkeley National Laboratory, and Computing and Communication Center at the Kyushu University, for use of the Earth Simulator, Hitachi SR8000/MPP, IBM SP-3 and IBM p5-model 595, respectively.

References

1. ASCI: http://www.llnl.gov/asci/
2. Earth Simulator Center: http://www.es.jamstec.go.jp/
3. Rabenseifner, R., Communication Bandwidth of Parallel Programming Models on Hybrid Architectures, Lecture Notes in Computer Science 2327 (WOMPEI 2002), 437-448, 2002.
4. Nakajima, K., Parallel Iterative Solvers of GeoFEM with Selective Blocking Preconditioning for Nonlinear Contact Problems on the Earth Simulator, ACM/IEEE Proceedings of SC2003, 2003.
5. Nakajima, K., Preconditioned Iterative Linear Solvers for Unstructured Grids on the Earth Simulator", IEEE Proceedings of HPC Asia 2004, 150-169, 2004.
6. GeoFEM: http://geofem.tokyo.rist.or.jp/
7. Saad, Y., Iterative Methods for Sparse Linear Systems, PWS Publishing Company, 1996.
8. Doi, S. and Washio, T., Using Multicolor Ordering with Many Colors to Strike a Better Balance between Parallelism and Convergence, Proceedings of RIKEN Symposium on Linear Algebra and its Applications, 19-26, 1999.
9. Kerbyson, D.J., Hoisie, A. and Wasserman, H., A Comparison Between the Earth Simulator and AlphaServer Systems using Predictive Application Performance Models, LA-UR-02-5222, Los Alamos National Laboratory, 2002.
10. Information Technology Center, The University of Tokyo: http://www.cc.u-tokyo.ac.jp/
11. National Energy Research Scientific Computing Center, Lawrence Berkeley National Laboratory: http://www.nersc.gov/
12. Computing and Communication Center, Kyushu University: http://www.cc.kyushu-u.ac.jp/
13. Adams, M.F., Bayraktar, H.H., Keaveny, T.M. and Papadopoulos, P., Applications of Algebraic Multigrid to Large-Scale Finite Element Analysis of Whole Bone Micro-Mechanics on the IBM SP, ACM/IEEE Proceedings of SC2003, 2003.
14. Oliker, L., Canning, A., Carter, J., Shalf, J., and Ethier, S., Scientific Computations on Modern Parallel Vector Systems, ACM/IEEE Proceedings of SC2004, 2004.
15. Uehara, H., Tamura, M., Itakura, K. and Yokokawa, M., MPI Performance Evaluation on the Earth Simulator (in Japanese), IPSJ Transactions on High-Performance Computing System, 44 SIG 1 (HPS 6), 24-34, 2003.
16. HLRS (High Performance Computing Center Stuttgart): http://www.hlrs.de/
17. Nakajima, K. and Okuda, H., Parallel Iterative Solvers with Selective Blocking Preconditioning for Simulations of Fault Zone Contact", Journal of Numerical Algebra with Applications, 11, 831-852, 2004.
18. Hatazaki, T., Lessons from porting vector computer applications onto Non-Uniform Memory Access scalar machines, IEEE Proceedings of HPC Asia 2004, 236-243, 2004.
19. Nakajima, K., Three-Level Hybrid vs. Flat MPI on the Earth Simulator: Parallel Iterative Solvers for Finite-Element Method, Applied Numerical Mathematics, 54, 237-255, 2005.
20. Nakajima, K., Parallel programming models for finite-element method using preconditioned iterative solvers with multicolor ordering on various types of SMP cluster supercomputers, IEEE Proceedings of HPC Asia 2005, 83-90, 2005.