Flat MPI and Hybrid Parallel Programming Models for FEM Applications on SMP Cluster Architectures

Kengo Nakajima
The 21st Century Earth Science COE Program
Department of Earth & Planetary Science
The University of Tokyo

First International Workshop on OpenMP (IWOMP 05)
June 1-4, 2005, Eugene, Oregon, USA.
Overview

• Background
  – GeoFEM Project & Earth Simulator
  – Preconditioned Iterative Linear Solvers

• Optimization Strategy on the Earth Simulator
  – BIC(0)-CG for Simple 3D Linear Elastic Applications

• Effect of reordering on various types of architectures
  – Contact Problems
  – PGA (Pin-Grid Array)
  – Multigrid

• Summary & Future Works
GeoFEM: FY.1998-2002
http://geofem.tokyo.rist.or.jp/

- Parallel FEM platform for solid earth simulation.
  - parallel I/O, parallel linear solvers, parallel visualization
  - solid earth: earthquake, plate deformation, mantle/core convection, etc.
- Part of national project by STA/MEXT for large-scale earth science simulations using the Earth Simulator.
- Strong collaborations between HPC and natural science (solid earth) communities.
- Current Activity
  - Research Consortium for Solid Earth Simulations on the Earth Simulator (9 sub-groups, >80 members)
System Configuration of GeoFEM

Utilities
- One-domain mesh
- Partitioner
- Partitioned mesh
- GPPView

Pluggable Analysis Modules
- Equation solvers
- Visualization (Vis.)
- Parallel I/O
- Communication (Comm.) I/F
- Solver I/F
- Structure
- Fluid
- Wave

Platform
- PEs
- Visualization data
Results on Solid Earth Simulation

Magnetic Field of the Earth: MHD code

Complicated Plate Model around Japan Islands

Simulation of Earthquake Generation Cycle in Southwestern Japan

Transportation by Groundwater Flow through Heterogeneous Porous Media

TSUNAMI!!

\[ \Delta h = 5.00 \]

\[ \Delta h = 1.25 \]

\[ T = 100 \quad T = 200 \quad T = 300 \quad T = 400 \quad T = 500 \]
Results by GeoFEM
Motivations

• GeoFEM Project (FY.1998-2002)

• Performance evaluation of FEM-type applications with complicated unstructured grids (not LINPACK, FDM ...) on the Earth Simulator (ES)
  – Implicit Linear Solvers
    • Preconditioned Iterative Solver
  – Hybrid vs. Flat MPI Parallel Programming Model
Earth Simulator (ES)
http://www.es.jamstec.go.jp/

- 640 × 8 = 5,120 Vector Processors
  - SMP Cluster-Type Architecture
  - 8 GFLOPS/PE
  - 64 GFLOPS/Node
  - 40 TFLOPS/ES
- 16 GB Memory/Node, 10 TB/ES
- 640 × 640 Crossbar Network
  - 12.3 GB/sec × 2
- Memory BWTH with 32 GB/sec.
- 35.6 TFLOPS for LINPACK (2002-March)
- 26 TFLOPS for AFES (Climate Simulation)
Flat MPI vs. Hybrid

**Flat-MPI** Each PE -> Independent

Hybrid Hierarchical Structure
Hybrid vs. Flat-MPI

• “Initial” Motivation for Hybrid
  – Block Jacobi-type Localized Parallel Preconditioning
  – Relatively lower convergence rate for many domains (or processors).
  – “Hybrid” may provide better convergence rate (i.e. faster convergence) because domain number is 1/8 of that of Flat-MPI programming model (on ES).
Local Data Structure
Node-based Partitioning
internal nodes - elements - external nodes
Block-Jacobi Type Localized ILU(0) Preconditioning [Nakajima et al. 1999]

- Global data dependency in ILU preconditioner is not suitable for parallel computing.

\[
y_k = b_k - \sum_{j=1}^{k-1} l_{kj} y_j \quad (k = 2, \cdots, N)
\]

\[
x_k = \tilde{d}_k \left( y_k - \sum_{j=k+1}^{N} u_{kj} y_j \right) \quad (k = N, N-1, \cdots, 1)
\]

- Ignore the effects by elements which belong to different partition / processor.

- Not as strong as original ILU.
  - How bad in many PE cases?
Block-Jacobi Type Localized ILU(0) Preconditioning [Nakajima et al. 1999]

Matrix components whose column numbers are outside the processor are ignored (set equal to 0) at Back-Forward Substitution.
Overlapped Additive Schwartz Domain Decomposition Method

Effect of additive Schwartz domain decomposition for solid mechanics example example with $3 \times 44^3$ DOFs on Hitachi SR2201, Number of ASDD cycle/iteration= 1, $\varepsilon = 10^{-8}$

<table>
<thead>
<tr>
<th>PE #</th>
<th>NO Additive Schwartz</th>
<th>WITH Additive Schwartz</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iter. #</td>
<td>Sec.</td>
</tr>
<tr>
<td>1</td>
<td>204</td>
<td>233.7</td>
</tr>
<tr>
<td>2</td>
<td>253</td>
<td>143.6</td>
</tr>
<tr>
<td>4</td>
<td>259</td>
<td>74.3</td>
</tr>
<tr>
<td>8</td>
<td>264</td>
<td>36.8</td>
</tr>
<tr>
<td>16</td>
<td>262</td>
<td>17.4</td>
</tr>
<tr>
<td>32</td>
<td>268</td>
<td>9.6</td>
</tr>
<tr>
<td>64</td>
<td>274</td>
<td>6.6</td>
</tr>
</tbody>
</table>
Overlapped Additive Schwartz Domain Decomposition Method

Effect of additive Schwartz domain decomposition for solid mechanics example example with $3 \times 44^3$ DOFs on Hitachi SR2201, Number of ASDD cycle/iteration= 1, $\varepsilon = 10^{-8}$

<table>
<thead>
<tr>
<th>PE #</th>
<th>NO Additive Schwartz</th>
<th>WITH Additive Schwartz</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sec.</td>
<td>Speed Up</td>
</tr>
<tr>
<td>1</td>
<td>204</td>
<td>233.7</td>
</tr>
<tr>
<td>2</td>
<td>253</td>
<td>143.6</td>
</tr>
<tr>
<td>4</td>
<td>259</td>
<td>74.3</td>
</tr>
<tr>
<td>8</td>
<td>264</td>
<td>36.8</td>
</tr>
<tr>
<td>16</td>
<td>262</td>
<td>17.4</td>
</tr>
<tr>
<td>32</td>
<td>268</td>
<td>9.6</td>
</tr>
<tr>
<td>64</td>
<td>274</td>
<td>6.6</td>
</tr>
</tbody>
</table>
Overlapped Additive Schwartz Domain Decomposition Method for Stabilizing Localized Preconditioning

**Global Operation**

\[ Mz = r \]

**Local Operation**

\[ M_{\Omega_1} z_{\Omega_1}^n = r_{\Omega_1}, \quad M_{\Omega_2} z_{\Omega_2}^n = r_{\Omega_2} \]

**Global Nesting Correction**

\[
\begin{align*}
z_{\Omega_1}^n &\leftarrow z_{\Omega_1}^{n-1} + M_{\Omega_1}^{-1} \left[ r_{\Omega_1} - M_{\Omega_1} z_{\Omega_1}^{n-1} - M_{\Gamma_{2\rightarrow1}} z_{\Gamma_{2\rightarrow1}}^{n-1} \right] \\
z_{\Omega_2}^n &\leftarrow z_{\Omega_2}^{n-1} + M_{\Omega_2}^{-1} \left[ r_{\Omega_2} - M_{\Omega_2} z_{\Omega_2}^{n-1} - M_{\Gamma_{1\rightarrow2}} z_{\Gamma_{1\rightarrow2}}^{n-1} \right]
\end{align*}
\]
Overlapped Additive Schwartz Domain Decomposition Method

Effect of additive Schwartz domain decomposition for solid mechanics example example with 3x44³ DOFs on Hitachi SR2201, Number of ASDD cycle/iteration= 1, $\varepsilon= 10^{-8}$

<table>
<thead>
<tr>
<th>PE #</th>
<th>NO Additive Schwartz</th>
<th>WITH Additive Schwartz</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iter. #</td>
<td>Sec.</td>
</tr>
<tr>
<td>1</td>
<td>204</td>
<td>233.7</td>
</tr>
<tr>
<td>2</td>
<td>253</td>
<td>143.6</td>
</tr>
<tr>
<td>4</td>
<td>259</td>
<td>74.3</td>
</tr>
<tr>
<td>8</td>
<td>264</td>
<td>36.8</td>
</tr>
<tr>
<td>16</td>
<td>262</td>
<td>17.4</td>
</tr>
<tr>
<td>32</td>
<td>268</td>
<td>9.6</td>
</tr>
<tr>
<td>64</td>
<td>274</td>
<td>6.6</td>
</tr>
</tbody>
</table>
Hybrid vs. Flat-MPI

• “Initial” Motivation for Hybrid
  – Block Jacobi-type Localized Parallel Preconditioning
  – Relatively lower convergence rate for many domains (or processors).
  – “Hybrid” may provide better convergence rate (i.e. faster convergence) because domain number is 1/8 of that of Flat-MPI programming model (on ES).

• Factors for Performance
  – Type of Applications, Problem Size
  – **Balance** of H/W Parameters

---

• Background
  – GeoFEM Project & Earth Simulator
  – Preconditioned Iterative Linear Solvers

• Optimization Strategy on the Earth Simulator
  – BIC(0)-CG for Simple 3D Linear Elastic Applications

• Effect of reordering on various types of architectures
  – PGA (Pin-Grid Array)
  – Contact Problems
  – Multigrid

• Summary & Future Works
Block IC(0)-CG Solver on the Earth Simulator

• 3D Linear Elastic Problems (SPD)
• Parallel Iterative Linear Solver
  – Node-based Local Data Structure
  – Conjugate-Gradient Method (CG): SPD
  – Localized Block IC(0) Preconditioning (Block Jacobi)
  – Additive Schwartz Domain Decomposition (ASDD)
  – http://geofem.tokyo.rist.or.jp/
• Hybrid Parallel Programming Model
  – OpenMP + MPI
  – Re-Ordering for Vector/Parallel Performance
  – Comparison with Flat MPI
Flat MPI vs. Hybrid

**Flat-MPI**: Each PE -> Independent

**Hybrid Hierarchical Structure**
Local Data Structure

Node-based Partitioning

internal nodes - elements - external nodes
1 SMP node => 1 domain for Hybrid Programming Model

MPI communication among domains
Basic Strategy for Parallel Programming on the Earth Simulator

• Hypothesis
  – Explicit ordering is required for unstructured grids in order to achieve higher performance in factorization processes on SMP node and vector processors.

\[
M = (L+D)D^{-1}(D+U) \\
Mz = r \\
D^{-1}(D+U) = z_1 \\
\text{Forward Substitution} \\
(L+D)z_1 = r : z_1 = D^{-1}(r-Lz_1) \\
\text{Backward Substitution} \\
(I+D^{-1}U)z_{\text{NEW}} = z_1 \\
z = z - D^{-1}Uz
\]
Basic Strategy for Parallel Programming on the Earth Simulator

- **Hypothesis**
  - Explicit ordering is required for unstructured grids in order to achieve higher performance in factorization processes on SMP node and vector processors.

Forward Substitution

\[(L+D)z = r : z = D^{-1}(r-Lz)\]

- do i = 1, N
  - WVAL = R(i)
  - do j = 1, INL(i)
    - WVAL = WVAL - AL(i,j) * Z(IAL(i,j))
  - enddo
  - Z(i) = WVAL / D(i)
- enddo

Backward Substitution

\[(I+ D I^{-1} U)z_{new} = z_{old} : z = z - D^{-1}Uz\]

- do i = N, 1, -1
  - SW = 0.0d0
  - do j = 1, INU(i)
    - SW = SW + AU(i,j) * Z(IAU(i,j))
  - enddo
  - Z(i) = Z(i) - SW / D(i)
- enddo

Dependency...

You need the most recent value of “z” of connected nodes. Vectorization/parallelization is difficult.

Reordering:

Directly connected nodes do not appear in RHS.
Basic Strategy for Parallel Programming on the Earth Simulator

• Hypothesis
  – Explicit ordering is required for unstructured grids in order to achieve higher performance in factorization processes on SMP node and vector processors.

• Re-Ordering for Highly Parallel/Vector Performance
  – Local operation and no global dependency
  – Continuous memory access
  – Sufficiently long loops for vectorization
Basic Strategy for Parallel Programming on the Earth Simulator

• Hypothesis
  – Explicit ordering is required for unstructured grids in order to achieve higher performance in factorization processes on SMP node and vector processors.

• Re-Ordering for Highly Parallel/Vector Performance
  – Local operation and no global dependency
  – Continuous memory access
  – Sufficiently long loops for vectorization

• 3-Way Parallelism for Hybrid Parallel Programming
  – Inter Node : MPI
  – Intra Node : OpenMP
  – Individual PE : Vectorization
Re-Ordering Technique for Vector/Parallel Architectures

Cyclic DJDS (RCM+CMC) Re-Ordering
(Doi, Washio, Osoda and Maruyama (NEC))

1. RCM (Reverse Cuthill-McKee)
2. CMC (Cyclic Multicolor)
3. DJDS re-ordering
4. Cyclic DJDS for SMP unit

These processes can be substituted by traditional multi-coloring (MC).
Re-Ordering Technique for Vector/Parallel Architectures
Cyclic DJDS(RCM+CMC) Re-Ordering
(Doi, Washio, Osoda and Maruyama (NEC))

K. Nakajima, “Parallel Iterative Solvers of GeoFEM with Selective Blocking Preconditioning for Nonlinear Contact Problems on the Earth Simulator“

Reordering = Coloring

- **COLOR**: Unit of independent sets.
- Elements grouped in the same “color” are independent from each other, thus parallel/vector operation is possible.
- Many colors provide faster convergence, but shorter vector length.: **Trade-Off !!**

![Red-Black (2 colors)](image1)

![4 colors](image2)

![Multicolor](image3)
“Incompatible nodes” are not affected by other nodes. System with fewer incompatible nodes provides faster convergence.
RCM Ordering
Red-Black Ordering
Highly Parallelized: Large Loop Length
Many Incompatible Nodes: Slow Convergence
Four-Color Ordering
Fairly Parallelized: Shorter Loop Length
Fewer Incompatible Nodes: Faster Convergence
Cyclic DJDS(RCM+CMC) for Forward/Backward Substitution in BILU Factorization

\[
\begin{align*}
\text{do } & \ iv = 1, \ \text{NCOLORS} \\
& \text{!$omp parallel do private (iv0,j,iS,iE,i,k,kk etc.)} \\
& \text{do } \ ip = 1, \ \text{PEsmpTOT} \\
& \quad \text{iv0} = \text{STACKmc}(\text{PEsmpTOT} \cdot (iv-1) + ip - 1) \\
& \quad \text{do } \ j = 1, \ \text{NLhyp}(iv) \\
& \quad \quad \text{iS} = \text{INL}(\text{npLX1} \cdot (iv-1) + \text{PEsmpTOT} \cdot (j-1) + ip - 1) \\
& \quad \quad \text{iE} = \text{INL}(\text{npLX1} \cdot (iv-1) + \text{PEsmpTOT} \cdot (j-1) + ip) \\
& \quad \text{!CDIR NODEP} \\
& \quad \text{do } \ i = \text{iv0} + 1, \ \text{iv0} + \text{iE} - \text{iS} \\
& \quad \quad k = i + \text{iS} - \text{iv0} \\
& \quad \quad \text{kk} = \text{IAL}(k) \\
& \quad \quad \text{(Important Computations)} \\
& \text{enddo} \\
& \text{enddo} \\
& \text{enddo} \\
& \text{enddo}
\end{align*}
\]
Simple 3D Cubic Model

Uniform Distributed Force in z-direction @ z=Z_{min}

\[ U_x = 0 \at x=X_{\text{min}} \]

\[ U_y = 0 \at y=Y_{\text{min}} \]

\[ U_z = 0 \at z=Z_{\text{min}} \]
Effect of Ordering
Effect of Re-Ordering

Long Loops
Continuous Access

PDJDS/CM-RCM
Proposed Method

Short Loops
Continuous Access

PDCRS/CM-RCM
Short innermost loop

Short Loops
Irregular Access

CRS no re-ordering
Effect of Re-Ordering

Results on 1 SMP node

Color #: 99 (fixed)

Re-Ordering is REALLY required !!!

Effect of Vector Length

X10 + Re-Ordering X100

22 GFLOPS, 34% of the Peak

Ideal Performance: 40%-45% for Single CPU
Effect of Re-Ordering

Results on 1 SMP node

Color #: 99 (fixed)

Re-Ordering is REALLY required !!!

80x80x80 case (1.5M DOF)

- 212 iter’s, 11.2 sec.
- 212 iter’s, 143.6 sec.
- 203 iter’s, 674.2 sec.
Ideal Performance: 48.6% of Peak

Ideal performance of the following MVP process for $3 \times 3$ block operation on the Earth Simulator (peak performance: 8 GFLOPS/PE, peak memory bandwidth: 32 GB/sec/PE) is estimated as 48.6% of the peak performance (3.89 GFLOPS).

```fortran
  do j= 1, m
    do i= 1, N
      k=(j-1)*N + i
      kk= index(k)
      Y(3*i-2)= Y(3*i-2) + A(9*k-8)*X(3*kk-2) &
                 + A(9*k-7)*X(3*kk-1) &
                 + A(9*k-6)*X(3*kk )
      Y(3*i-1)= Y(3*i-1) + A(9*k-5)*X(3*kk-2) &
                 + A(9*k-4)*X(3*kk-1) &
                 + A(9*k-3)*X(3*kk )
      Y(3*i )= Y(3*i ) + A(9*k-2)*X(3*kk-2) &
                 + A(9*k-1)*X(3*kk-1) &
                 + A(9*k )*X(3*kk )
    enddo
  enddo
```
SMP node # > 10
up to 176 nodes (1408 PEs)

Problem size for each SMP node is fixed.

PDJDS-CMC/RCM, Color #: 99
3D Elastic Model (Large Case)
256x128x128/SMP node, up to 2,214,592,512 DOF

GFLOPS rate

Parallel Work Ratio

3.8TFLOPS for 2.2G DOF
176 nodes (33.8% of peak)
3D Elastic Model (Small Case)
64x64x64/SMP node, up to 125,829,120 DOF

GFLOPS rate

Parallel Work Ratio

Flat MPI, Hybrid
3D Elastic Model
Problem Size and Parallel Performance
8-176 SMP nodes of the Earth Simulator
3D Elastic Model
Problem Size and Parallel Performance
8-176 SMP nodes of the Earth Simulator

Hybrid is faster
FlatMPI is faster

Large SMP Node #

Flat MPI, --- Hybrid
Hybrid outperforms Flat-MPI

• when ...
  – number of SMP node (PE) is large.
  – problem size/node is small.
• because flat-MPI has ...
  – as 8 times as many communication processes
  – as TWICE as large communication/computation ratio
• Effect of communication becomes significant if number of SMP node (or PE) is large.
• Performance Estimation by D. Kerbyson (LANL)
  – LA-UR-02-5222
  – relatively larger communication latency of ES
Comm. latency is relatively large in ES
D.J.Kerbyson et al. LA-UR-02-5222

<table>
<thead>
<tr>
<th></th>
<th>ES</th>
<th>ASCI-Q HP ES45</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Proc. Speed</strong></td>
<td>500 MHz</td>
<td>1.25 GHz</td>
</tr>
<tr>
<td><strong>SMP nodes</strong></td>
<td>8 x 640= 5,120</td>
<td>4 x 3,072=12,288</td>
</tr>
<tr>
<td><strong>Peak Speed</strong></td>
<td>8 x 5,120 = 40 TFLOPS</td>
<td>2.5 x 12,288 = 30 TFLOPS</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>2 x 5,120= 10 TB</td>
<td>4 x 12,288= 48 TB</td>
</tr>
<tr>
<td><strong>Peak Memory BWTH</strong></td>
<td>32 GB/s</td>
<td>2 GB/s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>L1/L2 20/30 GB/s</td>
</tr>
<tr>
<td><strong>Inter-Node MPI Comm.</strong></td>
<td>Latency: 5.6 μsec</td>
<td>Latency: 5 μsec</td>
</tr>
<tr>
<td></td>
<td>BWTH : 11.8 GB/s</td>
<td>BWTH: 300 MB/s</td>
</tr>
</tbody>
</table>
Performance Estimation for Finite-Element Type Application on ES using performance model (not real computations)

D.J. Kerbyson et al. LA-UR-02-5222 (LANL)
Hybrid outperforms Flat-MPI (cont.)

- Difference between Hybrid & Flat MPI of iteration number for convergence is not so significant … due to additive Schwartz domain decomposition.
Various Platforms

• Earth Simulator (ES)
  – Earth Simulator Center, JAMSTEC.
  – Vector Processors

• Hitachi SR8000
  – The University of Tokyo
  – Power-PC based Architecture
  – Pseudo-Vector Capability with Preload/Prefetch

• IBM SP-3
  – NERSC/Lawrence Berkeley National Laboratory: Seaborg
  – Power3
  – 8MB L2-cache for each PE
## Platforms

<table>
<thead>
<tr>
<th></th>
<th>ES</th>
<th>SP3</th>
<th>SR8k</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PE#/node</strong></td>
<td>8</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td><strong>Peak GF/PE</strong></td>
<td>8.0</td>
<td>1.5</td>
<td>1.8</td>
</tr>
<tr>
<td><strong>Mem.BW GB/s/node</strong></td>
<td>256</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td><strong>MPI lat. µsec</strong></td>
<td>5.6</td>
<td>16.3*</td>
<td>6-20**</td>
</tr>
<tr>
<td><strong>Network BW GB/s/Node</strong></td>
<td>12.3</td>
<td>1.00</td>
<td>1.60</td>
</tr>
</tbody>
</table>

*) Oliker et al.
**) HLRS
Reordering for the Earth Simulator
Matrix Storage for ILU Factorization

Long Loops Continuous Access
PDJDS/CM-RCM

Short Loops Continuous Access
PDCRS/CM-RCM
short innermost loop

Short Loops Irregular Access
CRS no re-ordering

Long Loops
Continuous Access

Short Loops
Continuous Access

Short Loops
Irregular Access
3D Elastic Simulation
Problem Size ~ GFLOPS

Earth Simulator
1 SMP node (8 PE’s)

Flat-MPI
23.4 GFLOPS, 36.6 % of Peak

OpenMP
21.9 GFLOPS, 34.3 % of Peak
3D Elastic Simulation
Problem Size~GFLOPS

Hitachi-SR8000-MPP with Pseudo Vectorization
1 SMP node (8 PE’s)

Flat-MPI
2.17 GFLOPS, 15.0 % of Peak

OpenMP
2.68 GFLOPS, 18.6 % of Peak
3D Elastic Simulation
Problem Size ~ GFLOPS
IBM-SP3 (NERSC)
1 SMP node (8 PE’s)

Cache is well-utilized in Flat-MPI.

Flat-MPI
OpenMP
# 3D Elastic Simulation

**Single PE Performance**

GFLOPS, 104 colors

<table>
<thead>
<tr>
<th>Earth Simulator</th>
<th>DJDS</th>
<th>CRS</th>
<th>DJDS</th>
<th>CRS</th>
<th>DJDS</th>
<th>CRS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hybrid</td>
<td>1.67</td>
<td>-</td>
<td>2.06</td>
<td>-</td>
<td>3.01</td>
<td>-</td>
</tr>
<tr>
<td>FlatMPI</td>
<td>2.75</td>
<td>-</td>
<td>3.02</td>
<td>-</td>
<td>3.19</td>
<td>-</td>
</tr>
<tr>
<td>Hitachi SR8000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hybrid</td>
<td>0.404</td>
<td>0.127</td>
<td>0.437</td>
<td>0.140</td>
<td>0.423</td>
<td>0.146</td>
</tr>
<tr>
<td>FlatMPI</td>
<td>0.366</td>
<td>0.128</td>
<td>0.343</td>
<td>0.138</td>
<td>0.322</td>
<td>0.146</td>
</tr>
<tr>
<td>IBM SP3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hybrid</td>
<td>0.145</td>
<td>0.167</td>
<td>0.142</td>
<td>0.165</td>
<td>0.137</td>
<td>0.145</td>
</tr>
<tr>
<td>FlatMPI</td>
<td>0.138</td>
<td>0.165</td>
<td>0.135</td>
<td>0.159</td>
<td>0.126</td>
<td>0.141</td>
</tr>
</tbody>
</table>
Performance of Intra-Node MPI

8 PE’s: Measurement using SEND/RECV subroutines in GeoFEM

- Elapsed time for 1,000 iterations (sec.)
- Elapsed time for 1,000 iterations without estimated latency (sec.)
- Estimated communication bandwidth (GB/sec)
Performance of Intra-Node MPI

8 PE’s: Measurement using inter-domain SEND/RECV subroutines in GeoFEM

<table>
<thead>
<tr>
<th></th>
<th>ES</th>
<th>SR8k</th>
<th>SP3</th>
</tr>
</thead>
<tbody>
<tr>
<td>PE#/node</td>
<td>8</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Peak GF/PE</td>
<td>8.0</td>
<td>1.8</td>
<td>1.5</td>
</tr>
<tr>
<td>Mem.BW GB/s/node</td>
<td>256</td>
<td>32</td>
<td>16</td>
</tr>
<tr>
<td>MPI lat. μsec</td>
<td>5.6</td>
<td>6-20**</td>
<td>16.3*</td>
</tr>
<tr>
<td>Network BW GB/s/Node</td>
<td>12.3</td>
<td>1.60</td>
<td>1.00</td>
</tr>
<tr>
<td>Intra-node MPI lat. μsec</td>
<td>5.29</td>
<td>9.17</td>
<td>8.32</td>
</tr>
<tr>
<td>Intra-node MPI BW MB/s/PE</td>
<td>1018.</td>
<td>95.8</td>
<td>52.5</td>
</tr>
</tbody>
</table>

Intra-node communication performance seems reasonable according to memory BW, inter-node communication performance.
Summary: Single Node Performance
Flat MPI vs. Hybrid

• Earth Simulator (ES)
  – Flat MPI is slightly better.

• Hitachi SR8000
  – Similar feature with that of ES
  – Hybrid is better for PDJDS/CM-RCM ordering.
  – Difference of single PE performance between Hybrid and Flat MPI for PDJDS/CM-RCM is significant.
  • compiler ?

• IBM SP-3
  – Effect of cache (8MB/PE) is significant.
  – Cache is well-utilized in Flat MPI
  – Performance in larger problems is similar
Speed-Up for Fixed Problem Size/PE

- Hybrid, □ Flat MPI (Lines: Ideal speed-up extrapolated from 1-node performance)

Earth Simulator

IBM SP-3 (Seaborg at NERSC)

- 12.6M DOF/node
- 2.2G DOF
- 3.80 TFLOPS
- 33.7% of peak
- (176 nodes)

- 3.0M DOF/node
- 0.38G DOF
- 110. GFLOPS
- 7.16% of peak
- (128 nodes)
- (8 PEs/node)
Speed-Up for Fixed Problem Size/PE

- Hybrid, □ Flat MPI (Lines: Ideal speed-up extrapolated from 1-node performance)

Earth Simulator

IBM SP-3 (Seaborg at NERSC)

![Graph showing speed-up performance for different systems.](image-url)
Speed-Up for Fixed Problem Size/PE

- Hybrid, Flat MPI (Lines: Ideal speed-up extrapolated from 1-node performance)

**Hitachi SR8000**

Graphs showing performance for different problem sizes:

- **786.4K DOF/node**
  - 0.10G DOF, 128 nodes
    - 271 GFLOPS, 14.7% of peak
    - 276 GFLOPS, 15.0%

- **6.29M DOF/node**
  - 8.05G DOF, 128 nodes
    - 335 GFLOPS, 18.2% of peak
    - 257 GFLOPS, 14.7%
## Platforms

<table>
<thead>
<tr>
<th></th>
<th>ES</th>
<th>SP3</th>
<th>SR8k</th>
</tr>
</thead>
<tbody>
<tr>
<td>PE#/node</td>
<td>8</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>Peak GF/PE</td>
<td>8.0</td>
<td>1.5</td>
<td>1.8</td>
</tr>
<tr>
<td>Mem.BW GB/s/node</td>
<td>256</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>MPI lat. μsec</td>
<td>5.6</td>
<td>16.3*</td>
<td>6-20**</td>
</tr>
<tr>
<td>Network BW GB/s/Node</td>
<td>12.3</td>
<td>1.00</td>
<td>1.60</td>
</tr>
</tbody>
</table>

**Comparison:**
- ES: SP3 : SR8k = 2.9 : 1 : 1.2-2.7
- Network BW = 12 : 1 : 1.6
Summary: Performance for Many Nodes, Flat MPI vs. Hybrid

• Earth Simulator (ES)
  – Effect of MPI latency is significant for:
    • Flat MPI
    • Many node #
    • small problem size/node

• Hitachi SR8000

• IBM SP-3
  – Effect of MPI latency is hidden by their relatively low communication bandwidth.
  – Performance with 128 SMP nodes are identical with extrapolation of single node performance, especially if problem size/node is sufficiently large.
• Background
  – GeoFEM Project & Earth Simulator
  – Preconditioned Iterative Linear Solvers

• Optimization Strategy on the Earth Simulator
  – BIC(0)-CG for Simple 3D Linear Elastic Applications

• **Effect of reordering on various types of architectures**
  – PGA (Pin-Grid Array)
  – Contact Problems
  – Multigrid

• Summary & Future Works
Elastic Problems
with Cubic Model
Effect of Loop Length (Color#)

100^3 nodes = 3 \times 10^6 DOF

PDCRS/CM-RCM Re-Ordering

Iterations for Convergence

![Graph showing the effect of loop length on iterations for convergence.]
Effect of Loop Length (Color#)

$100^3$ nodes = $3 \times 10^6$ DOF
PDJDS/CM-RCM Re-Ordering
Earth Simulator (single node)

- Flat MPI
- Hybrid

Graphs showing performance metrics (GFLOPS and seconds) vs. number of colors.
Effect of Loop Length (Color#)

100^3\nodes = 3 \times 10^6 \DOF
PDJDS/CM-RCM Re-Ordering
Earth Simulator (single node)

\[ \text{Smaller Vector Length} \]

\begin{align*}
\text{GFLOPS} \\
\text{sec.}
\end{align*}

\begin{align*}
\text{COLORS} \\
\text{COLORS}
\end{align*}

\begin{align*}
\bullet \text{ Flat MPI} \\
\bigcirc \text{ Hybrid}
\end{align*}
Effect of Loop Length (Color#)

$100^3$ nodes = $3 \times 10^6$ DOF

PDJDS/CM-RCM Re-Ordering

Earth Simulator (single node)

- Flat MPI
- Hybrid

Smaller Vector Length + OpenMP Overhead
Many Colors ... Trade-Off

- Fast convergence according to iteration number
  - S. Doi et al.
- Smaller vector length
  - FLOPS rate decreases.

- Hybrid is much more sensitive to color number
  - Especially on the Earth Simulator
Hybrid is much more sensitive to color numbers!

```fortran
!
$omp parallel do private (iv0,j,iS,iE,i,k,kk etc.)
!
!
!CDIR NODEP
!
!
!
!$omp parallel do private (iv0,j,iS,iE,i,k,kk etc.)
!
!
!CDIR NODEP
!
!
!
!CDIR NODEP
!
!$omp parallel do private (iv0,j,iS,iE,i,k,kk etc.)
!
!
!
!$omp parallel do private (iv0,j,iS,iE,i,k,kk etc.)
!
!
!
!$omp parallel do private (iv0,j,iS,iE,i,k,kk etc.)
!
!
!
!$omp parallel do private (iv0,j,iS,iE,i,k,kk etc.)
!
!
!$omp parallel do private (iv0,j,iS,iE,i,k,kk etc.)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
!
!
!(Important Computations)
Effect of Loop Length (Color#)

100³ nodes = 3x10⁶ DOF

PDJDS/CM-RCM Re-Ordering

Hitachi SR8k (single node)

Performance is not so sensitive to color# as the Earth Simulator. In Flat-MPI, performance is rather improved.
Effect of Loop Length (Color#)

100^3 nodes = 3 \times 10^6 DOF
PDCRS/CM-RCM Re-Ordering
IBM SP3 (single node, 8PE’s)

Performance is not so sensitive to color# as the Earth Simulator.

![Graphs showing performance vs. colors for different setups](image)
Effect of Loop Length (Color#)

$100^3$ nodes = $3 \times 10^6$ DOF

IBM SP3 (single node, 8PE’s)

- Flat MPI
- Hybrid

```
FLOPS
0.00 0.50 1.00 1.50
10 100 1000
```

```
COLORS
```

```
COLORS
```

```
COLORS
```

```
COLORS
```

```
COLORS
```

```
COLORS
```

```
COLORS
```
South-West Japan Model Contact Problems

Selective Blocking Preconditioning
South-West Japan
South-West Japan Model
800K nodes, 2.4M DOF
1-SMP nodes (8 PE’s)
South-West Japan Model
800K nodes, 2.4M DOF
ES, 1-SMP nodes (8 PE’s)
South-West Japan Model
800K nodes, 2.4M DOF
Hitachi SR8000, 1-SMP nodes (8 PEs)

- Flat MPI
- Hybrid (OpenMP)

![Graphs showing performance metrics vs. number of colors.](image)
South-West Japan Model
800K nodes, 2.4M DOF
Hitachi SR8000, 1-SMP nodes (8 PE's)

Flat MPI
Hybrid (OpenMP)
South-West Japan Model
800K nodes, 2.4M DOF
Hitachi SR8000, 1-SMP nodes (8 PE's)

- Flat MPI
- Hybrid (OpenMP)
South-West Japan Model
800K nodes, 2.4M DOF
Hitachi SR8000, 1-SMP nodes (8 PEs)

WHY? WHY?
South-West Japan Model
800K nodes, 2.4M DOF
IBM SP3, 1-SMP nodes (8 PE’s)
South-West Japan Model
800K nodes, 2.4M DOF
IBM SP3, 1-SMP nodes (8 PE’s)
South-West Japan Model
7,767,002 nodes, 23,301,006 DOF
ES, 10 SMP nodes (Peak= 640 GFLOPS)
South-West Japan Model
7,767,002 nodes, 23,301,006 DOF
ES, 10 SMP nodes (Peak= 640 GFLOPS)
Complicated Geometries
PGA for Mobile Pentium III
PGA model
61 pins, 956,128 elem, 1,012,354 nodes
(3,037,062 DOF)
PGA model
61 pins, 956,128 elem, 1,012,354 nodes
(3,037,062 DOF), 1 SMP node
PDJDS/MC Re-Ordering

![Graph](image-url)

- Flat MPI
- OpenMP
PGA model
61 pins, 956,128 elem, 1,012,354 nodes (3,037,062 DOF), 1 SMP node
PDJDS/MC Re-Ordering, Earth Simulator

Smaller Vector Length

Smaller Vector Length + OpenMP Overhead

GFLOPS

sec.

Colors

Flat MPI
OpenMP

IWOMP, JUN05
PGA model
61 pins, 956,128 elem, 1,012,354 nodes (3,037,062 DOF), 1 SMP node
PDJDS/MC Re-Ordering, Hitachi SR8000

Why? Why?
PGA model
61 pins, 956,128 elem, 1,012,354 nodes (3,037,062 DOF), 1 SMP node
PDJDS/MC Re-Ordering, IBM-SP3

Flat MPI
OpenMP
PGA model
61 pins, 956,128 elem, 1,012,354 nodes
(3,037,062 DOF) , 1 SMP node
Hitachi SR8000

Flat MPI
OpenMP

WHY ?
Single PE test for simple cubic model

786,432 DOF = 3x64^3 nodes

Hitachi SR8000

Behavior of Flat MPI with PDJDS seem like that of scalar processors, while OpenMP/PDJDS looks like vector processors.

Feature (or problem) of compiler? Pseudo-Vector option does not work well in Flat MPI.
PGA model
61 pins, 956,128 elem, 1,012,354 nodes (3,037,062 DOF), 1 SMP node
IBM-SP3

Flat MPI
OpenMP
PDCRS is more suitable for scalar processors

- Reduction type loop of CRS is more suitable for cache-based scalar processor because of its localized operation.
- PDJDS provides data locality if the color number increases.
  - On scalar processors, performance may be improved as color number increases.
PGA model
61 pins, 956,128 elem, 1,012,354 nodes (3,037,062 DOF) , 1 SMP node
IBM-SP3

Performance is slightly improved due to data locality.
PGA model
61 pins, 956,128 elem, 1,012,354 nodes (3,037,062 DOF), 1 SMP node

IBM-SP3

Performance is slightly improved due to data locality.
PGA model
61 pins, 956,128 elem, 1,012,354 nodes (3,037,062 DOF), 1 SMP node
IBM-SP3

PDJDS/MC

PDCRS/MC

GFLOPS

GFLOPS

OpenMP overhead

OpenMP overhead

Flat MPI

OpenMP
PGA model
61 pins, 956,128 elem, 1,012,354 nodes (3,037,062 DOF), 1 SMP node
Itanium II (SGI/Altix) (8 PE’s)
PGA model
61 pins, 956,128 elem, 1,012,354 nodes
(3,037,062 DOF), 1 SMP node
**AMD Opteron (8 PE’s)**

![Graphs showing performance comparison]

- Flat MPI (PDJDS)
- Flat MPI (PDCRS)
Multigrids
Multigrid

Multigrid is scalable !!!

Based on LLNL
Target Application
Thermal Convection between 2 Spherical Surfaces

• Typical Geometry in Earth Science Simulations
  – Mantle, Core etc.
  – Also applicable to external flow

• Boundary Conditions
  – \( r = R_{\text{in}} \)
    • \( u=v=w=0, \ T=1 \)
  – \( r = R_{\text{out}} \)
    • \( u=v=w=0, \ T=0 \)
  – Heat Generation

\( R_{\text{in}} = 0.50 \)
\( R_{\text{out}} = 1.00 \)
Semi-Implicit Pressure Correction Scheme

• Governing equations
  – Momentum (Navier-Stoke)
  – Continuity
  – Energy

• Poisson equations for pressure correction derived from continuity constraint.

• Parallel MGCG iterative method for Poisson eqn’s
  – V-cycle, Geometrical Multigrid
  – Gauss-Seidel Smoothing
  – Semi-coarsening
  – Multilevel communication table
Start from Icosahedron
Surface Grid by Adaptive Mesh Refinement (AMR)

Level 0
12 nodes
20 tri's

Level 1
42 nodes
80 tri's

Level 2
162 nodes
320 tri's

Level 3
642 nodes
1,280 tri's

Level 4
2,562 nodes
5,120 tri's
Semi-Unstructured Prismatic Grids

- generated from unstructured surface triangles
- structured in normal-to-surface direction
- flexible
- suitable for near-wall boundary layer computation
Results on Hitachi SR2201

320x900=288,000 cells/PE
up to 37M DOF on 128 PEs

- ICCG
- MGCG/FGS
- MGCG/ILUs
How about on ES?

• 1 CPU
  – 768,000 elements (1280 x 600)
  – ICCG: 29 sec. (Xeon 2.8GHz) → 169 sec. (ES)
  – MGCG: 58 sec. (Xeon 2.8GHz) → 345 sec. (ES)

• Why?
  – CRS: loop length is small

• Remedy
  – Reordering done in BILU(0)
    • Gauss-Seidel smoothing: dependency
  – Traditional Multicoloring
Critical Issue of Multigrid for Vector Computer

- Shorter Loop-Length for Coarse Grid Level.
Vectorization : Re-Ordering
Earth Simulator

- RCM (Reverse Cuthil-McKee), Multicolor

768,000 elem's
ES, 1 PE

MGCG (MC): 345 sec.
ICCG (original): 169 sec.
Many colors...
- fewer iterations for convergence
- lower performance due to shorter loops (especially in MG)
Scalability: Flat MPI
768,000 elem's/PE, up to 80 PE's

\[
\text{sec. vs. DOF}
\]

- MGCG (RCM)
- ICCG (MC=1500)
Next step: Hybrid

- Many colors are required for reasonably fast convergence.
- Overhead of OpenMP.

```plaintext
!$omp parallel do private (iv0,j,iS,iE,i,k,kk etc.)
do iv= 1, NCOLORS
  !$omp parallel do private (iv0,j,iS,iE,i,k,kk 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)
  (Important Computations)
endo
do
do
do
do
endo
do
do
endo
endo
```
Hybrid vs. Flat MPI
8PEs on ES (1 SMP node)

MGCG (10 iterations)

ICCG (100 iterations)
MGCG on ES

- Excellent performance by reordering
  - More than 20% of peak even in MGCG
    - Nice for Poisson solvers
    - 75% of ICCG
- Excellent scalability for flat-MPI
- Hybrid
  - Low performance for many colors
    - Significant for MGCG
  - But we need many colors for reasonably fast convergence
  - Not recommended
Hybrid vs. Flat MPI
8PE’s on ES (1 SMP node)

1280x600x8= 6,144,000 elements

MGCG (10 iterations)

ICCG (100 iterations)
Hybrid vs. Flat MPI
8PE’s on Hitachi SR8000
(1 SMP node)

1280x600x8 = 6,144,000 elements

MGCG (10 iterations)  

ICCG (100 iterations)
Hybrid vs. Flat MPI
8PE’s on IBM-SP3
(1 SMP node)

1280x600x8 = 6,144,000 elements

MGCG (10 iterations)

ICCG (100 iterations)
• Background
  – GeoFEM Project & Earth Simulator
  – Preconditioned Iterative Linear Solvers
• Optimization Strategy on the Earth Simulator
  – BIC(0)-CG for Simple 3D Linear Elastic Applications
• Effect of reordering on various types of architectures
  – PGA (Pin-Grid Array)
  – Contact Problems
  – Multigrid
• **Summary & Future Works**
Summary (Earth Simulator)

- Hybrid Parallel Programming Model on SMP Cluster Architecture with Vector Processors for Unstructured Grids
  - Nice parallel performance for both inter/intra SMP node on ES, 3.8TFLOPS for 2.2G DOF on 176 nodes (33.8%)
  - Re-Ordering is really required
  - Simple multicoloring is better for complicated geometries.
Summary (cont.) (Earth Simulator)

- Hybrid vs. Flat MPI
  - Flat-MPI is better for small number of SMP nodes.
  - Hybrid is better for large number of SMP nodes: Especially when problem size is rather small.
  - **Flat MPI: Communication, Hybrid: Memory**
  - depends on application, problem size etc.
  - Hybrid is much more sensitive to color numbers than flat MPI due to synchronization overhead of OpenMP, especially on the Earth Simulator.
    - This is not so significant in Hitachi SR8k & IBM SP3
Summary (cont.) (General)

- Hybrid vs. Flat MPI
  - In IBM SP-3, difference between flat-MPI and hybrid is not so significant, although flat-MPI is slightly better.
  - In Hitachi SR8000, pseudo-vector works better for hybrid cases.

- Effect of Color
  - In scalar processors (especially for flat-MPI), performance is getting better as color number increases, due to efficient utilization of cache.
Hybrid vs. Flat MPI

- Generally speaking, they are competitive.
  - Flat-MPI is better for multigrid type applications with multicolor ordering.
  - In ill-conditioned problems, Flat-MPI may require more iterations.
  - Depends on implementation of compiler
Hybrid vs. Flat MPI (cont.)

- Earth Simulator
  - Flat MPI is better
  - Hybrid is better for large PE number with small problem size.

- IBM SP-3
  - Flat MPI is better for small problem size because cache is utilized more efficiently

- Hitachi SR8000
  - Hybrid is better (mainly because of feature of compiler)

- **Careful treatment on color number is required for Hybrid programming model.**
My current feeling is ...

- Flat MPI is slightly better on the Earth Simulator in FEM-type applications with multicoloring.
Can Hybrid survive?
SAI (Sparse Approximate Inverse) preconditioning

- Suitable for parallel & vector computation.
  - especially for Hybrid Parallel Programming Model
  - only Mat-Vect. product.
  - reordering for avoiding dependency is not required.
    - corresponds to single-color

- Preconditioning for contact problems.
  - J.Zhang, K.Nakajima et al. (2003) for scalar processors
  - K.Nakajima (2005) for ES at WOMPEI05
 Matrix-Vector Product ONLY

!OMP PARALLEL DO
  do ip= 1, PEsmpTOT
    iv0= STACKmc(ip)
    do j= 1, NONdiagTOT
      iS= indexS(ip, j)
      iE= indexE(ip, j)
      !CDIR NODEP
      do i= iv0+1, iv0+iE-iS
        (computations)
      enddo
    enddo
  enddo
!OMP END PARALLEL DO
Earth Simulator, Mat-Vec.
1 SMP node (64GFLOPS peak), 500 Iter's.
Hybrid is rather faster !!

![Graph showing performance comparison between Flat MPI and Hybrid methods.](image-url)
Results: ES (1/2)

Results: ES (2/2)


\[ \square: SB-BIC(0), \quad \□: SAI: Large \]
\[ \square: SB-BIC(0), \quad \□: SAI: Small \]
Large (2.47M DOF)
Small (0.35M DOF)
Results: ES (2/2)

Results: Hitachi SR8000 (1/2)

Comparison: ES vs. Hitachi SR8k

ES

SR8k
Results: Hitachi SR8000 (2/2)


- SB-BIC(0), SAI: Large
- SB-BIC(0), SAI: Small

Large (2.47M DOF)
Small (0.35M DOF)
Acknowledgements

● Sponsor
  ○ MEXT, Japan.

● Computational Resources
  ○ Earth Simulator Center, Japan.
  ○ NERSC/Lawrence Berkeley National Laboratory, USA.
    ○ Dr. Horst D. Simon
    ○ Dr. Esmond G. Ng
  ○ Information Technology Center, The University of Tokyo, Japan.

● Colleagues …
  ○ GeoFEM Project

● IWOMP Committee