Parallel Multistage Preconditioners by Hierarchical Interface Decomposition on “T2K Open Super Computer (Todai Combined Cluster)” with Hybrid Parallel Programming Models

Kengo Nakajima
Information Technology Center, The University of Tokyo
JAMSTEC (Japan Agency for Marine-Earth Science & Technology)
CREST/JST (Japan Science & Technology Agency)

Work-in-Progress Presentation (WP-7), IEEE Cluster 2008
September 30th, 2008, Tsukuba, Japan
Motivations

• Parallel Preconditioners for Ill-Conditioned Problems
• T2K Open Super Computer
• Hybrid vs. Flat MPI
Technical Issues of “Parallel” Preconditioners for Iterative Solvers

- Simple problems can easily converge by simple preconditioners (e.g. Localized Block Jacobi) with excellent parallel efficiency.
- Difficult (ill-conditioned) problems cannot easily converge
  - Effect of domain decomposition on convergence is significant
  - More domains, more iterations for Localized Block Jacobi
Technical Issues of “Parallel” Preconditioners for Iterative Solvers

- If domain boundaries are on “harder” elements, convergence is very bad.
Technical Issues of “Parallel” Preconditioners for Iterative Solvers

• Simple problems can easily converge by simple preconditioners (e.g. Localized Block Jacobi) with excellent parallel efficiency.

• Difficult (ill-conditioned) problems cannot easily converge
  – Effect of domain decomposition on convergence is significant
  – More domains, more iterations for Localized Block Jacobi

• Remedies
  – deep fill-ins, deep overlapping
  – HID: Hierarchical Interface Decomposition
    • [Henon & Saad 2007]
T2K/Tokyo

- “T2K Open Super Computer (Todai Combined Cluster)
  - by Hitachi
  - Total 952 nodes (15,232 cores), 141 TFLOPS peak
    - Each Node = 4x AMD Quadcore Opteron (Barcelona) (2.3GHz)
    - 16th in TOP500 (June 2008) (fastest in Japan)

- up to 32 nodes (512 cores) in this work
Flat MPI vs. Hybrid

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

**Hybrid:** Hierarchical Structure
Goal of this Study

- HID for Ill-Conditioned Problems on T2K/Tokyo
  - Parallel FEM Applications

- Hybrid vs. Flat MPI Parallel Programming Models

- Effect of NUMA Control
Outline

• HID (Hierarchical Interface Decomposition)
• Hybrid Parallel Programming Model
  – Reordering
• Preliminary Results
• Summary & Future Works
HID: Hierarchical Interface Decomposition [Henon & Saad 2007]

- Multilevel Domain Decomposition
  - Extension of Nested Dissection
- Non-overlapped Approach: Connectors, Separators
- Suitable for Parallel Preconditioning Method
Parallel ILU for each Connector at each LEVEL

- The unknowns are reordered according to their level numbers, from the lowest to highest.
- The block structure of the reordered matrix leads to natural parallelism if ILU/IC decompositions or forward/backward substitution processes are applied.
• HID (Hierarchical Interface Decomposition)
• Hybrid Parallel Programming Model
  – Reordering
• Preliminary Results
• Summary & Future Works
Parallel Preconditioned Iterative Solvers on an SMP/Multicore node by OpenMP

- DAXPY, SMVP, Dot Products
  - Easy
- Factorization, Forward/Backward Substitutions in Preconditioning Processes
  - Global dependency
  - Reordering for parallelism required: forming independent sets
  - Multicolor Ordering (MC), Reverse-Cuthill-McKee (RCM)
    - both for parallel/vector performance
Cyclic-Multicoloring + RCM (CM-RCM)

- **Multicoloring (MC)**
  - Efficient
  - NOT robust for ill-conditioned problems
- **RCM**
  - Robust
  - Load-Balancing is bad for OpenMP on SMP/Multicore Node
- **CM-RCM**
  - Robust
  - Efficient
  - Excellent Load-Balancing
Ordering Methods

RCM

MC (Color#=4)

CM-RCM (Color#=4)
Ordering Method: In this Work

- CM-RCM with 10 colors
- Ordering is applied to vertices in each “level” on each “domain”

- If “level >1”, number of vertices are relatively smaller.
  - Color # in CM-RCM is controlled so that number of vertices is more that 10 for each color.
LEVEL-COLOR-THREAD

Communications at Each Level: Forward Substitutions

```
!$omp parallel do private(ip,i,SW1,SW2,SW3,isL,ieL,j,k,X1,X2,X3)
  do ip= 1, PEsmpTOT
    do i = STACKmc(ip-1,ic,lev)+1, STACKmc(ip,ic,lev)
      SW1= WW(3*i-2,R); SW2= WW(3*i-1,R); SW3= WW(3*i  ,R)
      isL= INL(i-1)+1; ieL= INL(i)
      do j= isL, ieL
        k= IAL(j)
        X1= WW(3*k-2,R); X2= WW(3*k-1,R); X3= WW(3*k  ,R)
        SW1= SW1 - AL(9*j-8)*X1 - AL(9*j-7)*X2 - AL(9*j-6)*X3
        SW2= SW2 - AL(9*j-5)*X1 - AL(9*j-4)*X2 - AL(9*j-3)*X3
        SW3= SW3 - AL(9*j-2)*X1 - AL(9*j-1)*X2 - AL(9*j  )*X3
      enddo
    X1= SW1; X2= SW2; X3= SW3
    X2= X2 - ALU(9*i-5)*X1
    X3= X3 - ALU(9*i-2)*X1 - ALU(9*i-1)*X2
    X3= ALU(9*i  )* X3
    X2= ALU(9*i-4)*( X2 - ALU(9*i-3)*X3 )
    X1= ALU(9*i-8)*( X1 - ALU(9*i-6)*X3 - ALU(9*i-7)*X2)
    WW(3*i-2,R)= X1; WW(3*i-1,R)= X2; WW(3*i  ,R)= X3
  enddo
enddo
!$omp end parallel do
enddo
```

call SOLVER_SEND_RECV_3_LEV(lev,...): Communications using Hierarchical Comm. Tables.
```
• HID (Hierarchical Interface Decomposition)
• Hybrid Parallel Programming Model
  – Reordering
• Preliminary Results
• Summary & Future Works
Target Application

- 3D Elastic Problems with Heterogeneous Material Property
  - $E_{\text{max}} = 10^3$, $E_{\text{min}} = 10^{-3}$, $\nu = 0.25$
    - generated by “sequential Gauss” algorithm for geo-statistics [Deutsch & Journel, 1998]
  - $128^3$ tri-linear hexahedral elements, 6,291,456 DOF

- (SGS+GPBiCG) Iterative Solvers
  - Symmetric Gauss-Seidel
  - Original Block Jacobi, HID

- T2K/Tokyo
  - 512 cores (32 nodes)

- FORTARN90 (Hitachi) + MPI
  - Flat MPI, Hybrid (4x4, 8x2)

- Effect of NUMA Control
Flat MPI, Hybrid (4x4, 8x2)
# Effect of NUMA Policy

## Command line switches

<table>
<thead>
<tr>
<th>Policy ID</th>
<th>Command line switches</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>no command line switches</td>
</tr>
<tr>
<td>1</td>
<td><code>--cpunodebind=$SOCKET</code> <code>--interleave=all</code></td>
</tr>
<tr>
<td>2</td>
<td><code>--cpunodebind=$SOCKET</code> <code>--interleave=$SOCKET</code></td>
</tr>
<tr>
<td>3</td>
<td><code>--cpunodebind=$SOCKET</code> <code>--membind=$SOCKET</code></td>
</tr>
<tr>
<td>4</td>
<td><code>--cpunodebind=$SOCKET</code> <code>--localalloc</code></td>
</tr>
<tr>
<td>5</td>
<td><code>--localalloc</code></td>
</tr>
</tbody>
</table>

### 32 nodes, 512 cores

12.6K DOF/core
Effect of NUMA Policy: Best Cases

<table>
<thead>
<tr>
<th>Policy ID</th>
<th>command line switches</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>no command line switches</td>
</tr>
<tr>
<td>1</td>
<td>--cpunodebind=$SOCKET --interleave=all</td>
</tr>
<tr>
<td>2</td>
<td>--cpunodebind=$SOCKET --interleave=$SOCKET</td>
</tr>
<tr>
<td>3</td>
<td>--cpunodebind=$SOCKET --membind=$SOCKET</td>
</tr>
<tr>
<td>4</td>
<td>--cpunodebind=$SOCKET --localalloc</td>
</tr>
<tr>
<td>5</td>
<td>--localalloc</td>
</tr>
</tbody>
</table>

8 nodes, 128 cores
50.3K DOF/core

32 nodes, 512 cores
12.6K DOF/core

Programming Models

- Flat MPI (policy3)
- Hybrid 4x4 (policy3)
- Hybrid 8x2 (policy2)
Relative Performance
HID vs. Block Jacobi
normalized by Block Jacobi

![Graphs showing relative performance of HID vs. Block Jacobi in iterations and seconds, normalized by Block Jacobi.](image)
Scalability: HID vs. Block Jacobi

HID

Localized Block Jacobi
• HID (Hierarchical Interface Decomposition)
• Hybrid Parallel Programming Model
  – Reordering
• Preliminary Results
• **Summary & Future Works**
Summary & Future Works

• HID for Ill-Conditioned Problems on T2K/Tokyo
  – Hybrid/Flat MPI parallel programming model
  – CM-RCM reordering
  – comparison with Localized Block Jacobi

• Generally speaking, HID is better than Localized Block Jacobi
  – more robust, more efficient

• Hybrid 4x4 and Flat MPI are competitive
  – Effect of NUMA policy is significant (if size/core is large)

• Future Works
  – Effect of higher order of fill-ins for robust convergence
  – Nested dissection ordering on each node for hybrid programming model
Flat MPI, Hybrid (4x4, 8x2)