Experiences with MPI/OpenMP on Multicore Architectures

Ulrike Meier Yang
Collaborators: A. Baker, H. Gahvari, T. Gamblin, M. Schulz
Review of Algebraic Multigrid (AMG)

Solve $Ax = b$

**Setup Phase**

- Select coarse “grids”
- Define interpolation, $P^{(m)}$, $m=1,2,…$
- Define restriction, $R^{(m)} = (P^{(m)})^T$
- Define coarse-grid operators, $A^{(m+1)} = R^{(m)} A^{(m)} P^{(m)}$

**Solve Phase (level m)**

- Smooth $A^{(m)} u^m = f^m$
- Compute $r^m = f^m - A^{(m)} u^m$
- Restrict $r^{m+1} = R^{(m)} r^m$
- Solve $A^{(m+1)} e^{m+1} = r^{m+1}$

- Smooth $A^{(m)} u^m = f^m$
- Correct $u^m \leftarrow u^m + e^m$
- Interpolate $e^m = P^{(m)} e^{m+1}$
Approach for parallelizing multigrid is straightforward data decomposition

- Basic communication pattern is “nearest neighbor”
  - Relaxation, interpolation, & Galerkin not hard to implement
- Different neighbor processors on coarse grids
- Many idle processors on coarse grids (100K+ on BG/L)
  - Algorithms to take advantage have had limited success
Parallel coarse-grid selection in AMG can produce unwanted side effects

- Non-uniform grids can lead to increased operator complexity and poor convergence
- Operator “stencil growth” reduces parallel efficiency

- Currently no guaranteed ways to control complexity
- Can ameliorate with more aggressive coarsening
- Requires long-range interpolation approaches
New parallel coarsening and long-range interpolation methods are improving scalability

- Unstructured 3D problem with material discontinuities
- About 90K unknowns per processor on MCR (Linux cluster)
- AMG - GMRES(10)

![Diagram showing coarsening and interpolation improvements.](image)

- New coarsening → 2.7x faster!
- New interpolation → 4.5x faster!
Multicore cluster details (Hera):

- Individual 512 KB L2 cache for each core
- 2 MB L3 cache shared by 4 cores
- 4 sockets per node, 16 cores sharing the 32 GB main memory
- NUMA memory access
Machine specification – Hera - LLNL

- Multi-core/ multi-socket cluster
- 864 diskless nodes interconnected by DDR Infiniband
- AMD opteron quad core (2.3 GHz)
- Fat tree network
Distance 1 vs 2 interpolation on different computer architectures

- AMG-GMRES(10), 7pt 3D Laplace problem on a unit cube, using 50x50x25 points per processor
AMG-GMRES(10) on Hera, 7pt 3D Laplace problem on a unit cube, 100x100x100 grid points (16 cores) per node
Communication patterns – 128 procs
Distance 1 vs. 2 interpolation

- Setup
- Solve
Communication patterns, 8 nodes, 128 cores, setup phase
Solve phase

- Communication plots for 128 core problem:
Performance profile of AMG solve cycle for 64 MPI tasks on Hera (Vampir)

- Proc id
- Time
- Computation
- Idle time
- MPI calls
AMG cycle time with varying MPI tasks/node

More than 6 MPI tasks/node is not helpful….

- pressure on the internode communication network
- local memory pressure
- process can migrate between cores and sockets (tuning)
How are kernels threaded?

- matrices are distributed across $P$ processors in contiguous blocks of rows
- Kernels are threaded such that each thread works on a contiguous subset of the rows
- smoother: GS within each thread, Jacobi on thread boundaries

\[ A = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_p \end{pmatrix} \]

\[ A_p = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{pmatrix} \]
Performance Analysis of Algorithms – preliminary results

- Problem: 7pt 3D Laplace on a 96 x 96 x 96 grid, AMG-GMRES(10), average cycle times, speedup
Profile of AMG-GMRES(10) solve phase, 7pt 3D Laplace on a 96 x 96 x 96 grid

<table>
<thead>
<tr>
<th>Routine</th>
<th>Percentage of Solve phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matvec</td>
<td>58.2</td>
</tr>
<tr>
<td>Smoother</td>
<td>29.9</td>
</tr>
<tr>
<td>MatvecT</td>
<td>7.6</td>
</tr>
<tr>
<td>Inner Product</td>
<td>2.6</td>
</tr>
<tr>
<td>Axpy</td>
<td>1.1</td>
</tr>
</tbody>
</table>

- Most time intensive kernels: Matvec, smoother
- Similar performance behavior for kernels
MxV Performance 2-16 cores – OpenMP vs. MPI
7pt stencil

OpenMP

MPI

Matrix size

Mflops

Matrix size

Mflops

---

Lawrence Livermore National Laboratory

LLNL-PRES-578892
Improving AMG’s Performance using MCSUP (MultiCore SUPport library)

Default Execution: all memory tends to be allocated on a single memory module (processor socket) – contention!

Optimization: allocate data on the same module/socket as the thread that will access it - improves performance!

- MCSUp supports …
  - Determination of hardware configuration
  - User-defined high level association of memory to threads
  - Thread pinning
  - Limited impact on code: use of special routines to allocate/free memory
New NUMA utilities resulted in a 3.5X speedup for matvec with 16 OpenMP threads

- Significant improvement in threaded performance
- OpenMP and MPI single node performance now comparable
- Baseline useful for future improvements
New NUMA utilities resulted in a 3.5X speedup for matvec with 16 OpenMP threads

- Significant improvement in threaded performance
- OpenMP and MPI single node performance now comparable
- Baseline useful for future improvements
New NUMA utilities resulted in a 3.5X speedup for matvec with 16 OpenMP threads

- Significant improvement in threaded performance
- OpenMP and MPI single node performance now comparable
- Baseline useful for future improvements
Use of MCSup within AMG-GMRES(10)

- Improved performance compared to OpenMP
- Still worse than MPI
AMG-GMRES(10) on Hera, 7pt 3D Laplace problem on a unit cube, 100x100x100 grid points per node (16 procs per node)
Hybrid MPI/OpenMP

AMG Solve Cycle on Hera, 3456 Cores

3D 7-point Laplace model problem, 50 x 50 x 25 points/core
AMG-GMRES(10) on Hera, 7pt 3D Laplace problem on a unit cube, 100x100x100 grid points per node (16 procs per node)
AMG-GMRES(10) on Hera, 7pt 3D Laplace problem on a unit cube, 100x100x100 grid points per node (16 procs per node)
Blue Gene/P Solution – Intrepid Argonne National Laboratory

- 40 racks with 1024 compute nodes each
- Quad-core 850 MHz PowerPC 450 Processor
- 163,840 cores
- 2GB main memory per node shared by all cores
- 3D torus network - isolated dedicated partitions
AMG-GMRES(10) on Intrepid, 7pt 3D Laplace problem on a unit cube, 50x50x25 grid points per core (4 cores per node)

No. Iterations

<table>
<thead>
<tr>
<th></th>
<th>H1x4</th>
<th>H2x2</th>
<th>MPI</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>17</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>1024</td>
<td>20</td>
<td>20</td>
<td>22</td>
</tr>
<tr>
<td>8192</td>
<td>25</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>27648</td>
<td>29</td>
<td>27</td>
<td>28</td>
</tr>
<tr>
<td>65536</td>
<td>30</td>
<td>33</td>
<td>29</td>
</tr>
<tr>
<td>128000</td>
<td>37</td>
<td>36</td>
<td>36</td>
</tr>
</tbody>
</table>
AMG-GMRES(10) on Intrepid, 7pt 3D Laplace problem on a unit cube, 50x50x25 grid points per core (4 cores per node)
Cray XT-5 Jaguar-PF
Oak Ridge National Laboratory

- 18,688 nodes in 200 cabinets
- Two sockets with AMD Opteron Hex-Core processor per node
- 16 GB main memory per node
- Network: 3D torus/mesh with wrap-around links (torus-like) in two dimensions and without such links in the remaining dimension
- Compute Node Linux, limited set of services, but reduced noise ratio
- PGI’s C compilers (v.904) with and without OpenMP, OpenMP settings: -mp=nonuma, -mp=numa (preemptive thread pinning, localized memory allocations)
AMG-GMRES(10) on Jaguar, 7pt 3D Laplace problem on a unit cube, 50x50x30 grid points per core (12 cores per node)
Machine specification – RZuSeq – LLNL

- IBM BG/Q
- 512 nodes
- 16 cores per node
- Up to 4-way simultaneous multithreading per core
- 5D-torus network
AMG-GMRES(10) on RZuSeq, 7pt 3D Laplace problem on a unit cube, 100x100x100 grid points per node (16 procs per node)
AMG-GMRES(10) on RZuSeq, 7pt 3D Laplace problem on a unit cube, 100x100x100 grid points per node (16 procs per node)
Thank You!

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. It also used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-06CH11357, as well as resources of the National Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. These resources were made available via the Performance Evaluation and Analysis Consortium End Station, a Department of Energy INCITE project.