WARP3D Implementation of MKL Cluster PARDISO Solver

Jeremy Nicklas¹, Karen Tomko¹, Robert Dodds², Kevin Manalo³

¹Ohio Supercomputer Center
²University of Illinois
³Maryland Advanced Research Computing Center
Overview

• WARP3D Overview
• Experiments and results for MKL PARDISO driver
• Experiments and results with WARP3D
• Conclusion
WARP3D

• Open-source code for nonlinear analysis of 3D solids using finite elements
• Primary applications: design, safety, life extension in heavy energy production systems
• Code extensively developed for 20 years in university
• Linux, Windows, Mac OS using Intel software tools
• Works with the iterative solver HYPRE and direct solver MKL Cluster PARDISO
• ~175K lines of code (Fortran 90-2008)

www.warp3d.net → documentation, ready-to-run executables, verification/example suites
www.github.com/rhdodds/warp3d → source code
Parallel Architecture

- FE mesh partitioned into domains then blocks of elements
- Domain # = MPI rank, > 1 thread per rank
- # elements per block tuned to # vector registers & cache architecture
- A thread (OpenMP) processes all computations for entire block
- Vectorization within each thread runs inner loops on # elements per block
- Structure of Arrays (SoA) design

Cracked flange:
- 923K nodes
- 216K elements
- 2.7M nonlinear eqns
Background on MKL PARDISO

• MKL PARDISO is a Parallel, Direct Sparse Matrix Solver
  • Cluster MKL - hybrid MPI/OpenMP implementation
• MKL PARDISO can account for 80-95% of total run time in WARP3D for large models
• Primary impact for WARP3D is **factorization time**
  • Algorithm based on Level-3 BLAS and using a combination of left- and right-looking supernode techniques
  • Called by WARP3D thousands of times per simulation
  • Sparsity structure of the matrix is fixed for a given simulation, with coefficients changing over simulation time
MKL PARDISO Driver

- Model
  - 1.4M equations, symm. pos. definite
  - 10.5 TFlop for factorization
  - 20 GB memory used for factorization

- Benchmarking on
  - Intel Broadwell (OSC Owens)
  - Intel KNL (TACC Stampede 1.5)

- KNL performs as well as Broadwell when utilizing all the threads
Intel Broadwell Environment

- Ohio Supercomputer Center (Owens Cluster)
- Node Description
  - Dell PowerEdge C6320 two-socket servers
  - Intel Xeon E5-2680 v4 (Broadwell, 14 cores, 2.40 GHz)
  - 128 GB Memory
- Mellanox EDR (100 Gbps) Infiniband networking
  - MPI Fabric: DAPL
  - `_MPI_DAPL_TRANSATION_CACHE=0`*
- Uses Intel 17.0.2 and Intel MPI 2017.2

Intel KNL Environment

• Texas Advanced Computing Center (Stampede2 Cluster)

• Node Description
  • Single-socket servers
  • Intel Xeon Phi 7250 (KNL, 68 cores, 1.40 GHz)
  • 96 GB DDR4 Memory + 16 GB MCDRAM

• Intel Omni-Path (100 Gbps) networking
  • MPI Fabric: TMI

• Memory/Cluster Modes Benchmarked
  • Cache Quadrant Mode
  • Flat Quadrant Mode – prefer MCDRAM

• Uses Intel 17.0.4 and Intel MPI 17.0.3
Finite Element Model

- Large plate-like structure with multiple holes
- 924,999 nodes
- 218,240 elements
- 2.73M equations
  - 52.7 TFlop for factorization
  - ~62 GB memory used for factorization
- 20-node elements
- Tension loading
- 5 load(time) steps that cause moderate yielding
Single Node Performance

- PARDISO takes 95-99% of total run time in WARP3D
- Best Broadwell is 1.37x faster than best KNL timing
- 20 threads for Broadwell and 26 threads for KNL are converged to within 10% of best timings
- Unable to run single-node Cache Quadrant mode on KNL due to memory issue
Intel Broadwell MPI

- Cluster PARDISO takes 83-96% of total run time in WARP3D
- Observed 1.06x speedup on single node when using 4 MPI processes
- For >1 node, best performance using 1 MPI rank per socket (2 MPI rank / node)
- Little improvement beyond 8 nodes (4.1x speedup for 2 MPI rank / node)
Intel KNL MPI

- Cluster PARDISO takes 73-96% of total run time in WARP3D
- Memory issue observed for Cache mode on single node
- Cache mode is 1.56x faster than Flat mode for 2 nodes and quickly converge for more nodes
- Little improvement beyond 8 nodes (3.4x speedup for Flat mode)
Conclusions and Insights

- Broadwell performs 1.37x faster than KNL on a single node (Flat mode)
- Best parallel performance observed when running a single MPI process per socket for dual-socket Broadwell
- Cache mode outperforms Flat mode for multi-node, but rapidly converges as more nodes are added
- Need to investigate distributed assembly option in WARP3D for Cluster PARDISO

Supported by the Intel® Parallel Computing Center at the Ohio Supercomputer Center and Texas Advanced Advanced Computing Center.
HYPRE Comparison

- Benchmarked on Intel Broadwell (16 Nodes)
- Solvers
  - Cluster PARDISO
  - HYPRE - parallel assembly across nodes
  - HYPRE - element stiffnesses computed on ranks, moved to root, assembled, hypre called
- HYPRE performance improves with less strict tolerance even when number of global Newton iterations increases from 16 → 18
- Need to investigate conversion to CSR for HYPRE with parallel assembly off (took 109 seconds)
- Need to investigate distributed assembly option for Cluster PARDISO

<table>
<thead>
<tr>
<th>Tolerance</th>
<th>Total Walltime (s)</th>
<th>Eqn Solves</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>516</td>
<td>16</td>
</tr>
<tr>
<td>1.00E-5</td>
<td>722</td>
<td>16</td>
</tr>
<tr>
<td>1.00E-4</td>
<td>597</td>
<td>16</td>
</tr>
<tr>
<td>1.00E-3</td>
<td>484</td>
<td>16</td>
</tr>
<tr>
<td>1.00E-2</td>
<td>374</td>
<td>16</td>
</tr>
<tr>
<td>1.00E-1</td>
<td>323</td>
<td>18</td>
</tr>
<tr>
<td>1.00E-1</td>
<td>421</td>
<td>18</td>
</tr>
</tbody>
</table>