High-Performance Computing for Mechanical Simulations using ANSYS

Jeff Beisheim
ANSYS, Inc
HPC Defined

High Performance Computing (HPC) at ANSYS:

An ongoing effort designed to remove computing limitations from engineers who use computer aided engineering in all phases of design, analysis, and testing.

It is a hardware and software initiative!
Need for Speed

Large

Fast

Extend

Assemblies
CAD-to-mesh
Capture fidelity

Impact product design
Enable large models
Allow parametric studies

Modal
Nonlinear
Multiphysics
Dynamics
A History of HPC Performance

- **1980’s**: Vector Processing on Mainframes
- **1990**
  - Shared Memory Multiprocessing (SMP) available
  - Iterative PCG Solver introduced for large analyses
- **1994**
  - 64-bit large memory addressing
  - Distributed PCG solver
  - Distributed ANSYS (DMP) released
  - Distributed sparse solver
  - Variational Technology
  - Support for clusters using Windows HPC
- **1999 - 2000**
  - Optimized for multicore processors
  - Teraflop performance at 512 cores
- **2004**
  - 1st company to solve 100M structural DOF
- **2005 - 2007**
  - Distributed PCG solver
  - Distributed sparse solver
  - Variational Technology
  - Support for clusters using Windows HPC
- **2007 - 2009**
  - Optimized for multicore processors
  - Teraflop performance at 512 cores
- **2010**
  - GPU acceleration (single GPU; SMP)
- **2012**
  - GPU acceleration (multiple GPUs; DMP)
  - Teraflop performance on 512 cores

**HPC Capability and Throughput**

**ANSYS**
HPC Revolution

Recent advancements have revolutionized the computational speed available on the desktop

- Multi-core processors
  - Every core is really an independent processor
- Large amounts of RAM and SSDs
- GPUs
Parallel Processing – Hardware

• 2 Types of memory systems
  – Shared memory parallel (SMP) ← single box, workstation/server
  – Distributed memory parallel (DMP) ← multiple boxes, cluster
Parallel Processing – Software

• 2 Types of parallel processing for Mechanical APDL
  – Shared memory parallel (-np > 1)
    • First available in v4.3
    • Can only be used on single machine
  – Distributed memory parallel (-dis -np > 1)
    • First available in v6.0 with the DDS solver
    • Can be used on single machine or cluster

• GPU acceleration (-acc)
  • First available in v13.0 using NVIDIA GPUs
  • Supports using either single GPU or multiple GPUs
  • Can be used on single machine or cluster
Distributed ANSYS Design Requirements

• No limitation in simulation capability
  – Must support all features
  – Continually working to add more functionality with each release

• Reproducible and consistent results
  – Same answers achieved using 1 core or 100 cores
  – Same quality checks and testing are done as with SMP version
  – Uses the same code base as SMP version of ANSYS

• Support all major platforms
  – Most widely used processors, operating systems, and interconnects
  – Supports same platforms that SMP version supports
  – Uses latest versions of MPI software which support the latest interconnects
Distributed ANSYS Design

- **Distributed steps (-dis -np N)**
  - At start of first load step, decompose FEA model into N pieces (domains)
  - Each domain goes to a different core to be solved
  - Solution is not independent!!
    - Lots of communication required to achieve solution
    - Lots of synchronization required to keep all processes together
  - Each process writes its own sets of files (file0*, file1*, file2*,..., file[N-1]*)
  - Results are automatically combined at end of solution
    - Facilitates post-processing in /POST1, /POST26, or WorkBench
Distributed ANSYS Capabilities

- Static linear or nonlinear analyses
- Buckling analyses
- Modal analyses
- Harmonic response analyses using FULL method
- Transient response analyses using FULL method
- Single field structural and thermal analyses
- Low-frequency electromagnetic analysis
- High-frequency electromagnetic analysis
- Coupled-field analyses
- All widely used element types and materials
- Superelements (use pass)
- NLGEOM, SOLC, LNSRCH, AUTOTS, IC, INISTATE, ...
- Linear Perturbation
- Multiframe restarts
- Cyclic symmetry analyses
- User Programmable features (UPFs)

Wide variety of features & analysis capabilities are supported
Distributed ANSYS Equation Solvers

- **Sparse direct solver (default)**
  - Supports SMP, DMP, and GPU acceleration
  - Can handle all analysis types and options
  - Foundation for Block Lanczos, Unsymmetric, Damped, and QR damped eigensolvers

- **PCG iterative solver**
  - Supports SMP, DMP, and GPU acceleration
  - Symmetric, real-value matrices only (i.e., static/full transient)
  - Foundation for PCG Lanczos eigensolver

- **JCG/ICCG iterative solvers**
  - Supports SMP only
Distributed ANSYS Eigensolvers

- **Block Lanczos eigensolver (including QRdamp)**
  - Supports SMP and GPU acceleration
- **PCG Lanczos eigensolver**
  - Supports SMP, DMP, and GPU acceleration
  - Great for large models (>5 MDOF) with relatively few modes (< 50)
- **Supernode eigensolver**
  - Supports SMP only
  - Optimal choice when requesting hundreds or thousands of modes
- **Subspace eigensolver**
  - Supports SMP, DMP, and GPU acceleration
  - Currently only supports buckling analyses; “beta” for modal in R14.5
- **Unsymmetric/Damped eigensolvers**
  - Supports SMP, DMP, and GPU acceleration
Distributed ANSYS Benefits

• Better architecture
  – More computations performed in parallel → faster solution time

• Better speedups than SMP
  – Can achieve > 10x on 16 cores (try getting that with SMP!)
  – Can be used for jobs running on 1000+ CPU cores

• Can take advantage of resources on multiple machines
  – Memory usage and bandwidth scales
  – Disk (I/O) usage scales
  – *Whole new class of problems can be solved!*
Distributed ANSYS Performance

- Need fast interconnects to feed fast processors
  - Two main characteristics for each interconnect: latency and bandwidth
  - Distributed ANSYS is highly bandwidth bound

---

**DISTRIBUTED ANSYS STATISTICS**

- **Release:** 14.5    **Build:** UP20120802    **Platform:** LINUX x64
- **Date Run:** 08/09/2012    **Time:** 23:07
- **Processor Model:** Intel(R) Xeon(R) CPU E5-2690 0 @ 2.90GHz

**Total number of cores available:** 32
**Number of physical cores available:** 32
**Number of cores requested:** 4 (Distributed Memory Parallel)
**MPI Type:** INTELMPI

<table>
<thead>
<tr>
<th>Core</th>
<th>Machine Name</th>
<th>Working Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>hpclnxsmc00</td>
<td>/data1/ansyswork</td>
</tr>
<tr>
<td>1</td>
<td>hpclnxsmc00</td>
<td>/data1/ansyswork</td>
</tr>
<tr>
<td>2</td>
<td>hpclnxsmc01</td>
<td>/data1/ansyswork</td>
</tr>
<tr>
<td>3</td>
<td>hpclnxsmc01</td>
<td>/data1/ansyswork</td>
</tr>
</tbody>
</table>

**Latency time from master to core:**
- 1 = 1.171 microseconds
- 2 = 2.251 microseconds
- 3 = 2.225 microseconds

**Communication speed from master to core:**
- 1 = 7934.49 MB/sec  \(\text{Same machine}\)
- 2 = 3011.09 MB/sec  \(\text{QDR Infiniband}\)
- 3 = 3235.00 MB/sec  \(\text{QDR Infiniband}\)
Distributed ANSYS Performance

- Need fast interconnects to feed fast processors

- Turbine model
- 2.1 million DOF
- SOLID187 elements
- Nonlinear static analysis
- Sparse solver (DMP)
- Linux cluster (8 cores per node)
Distributed ANSYS Performance

• Need fast hard drives to feed fast processors
  – Check the bandwidth specs
    – ANSYS Mechanical can be highly I/O bandwidth bound
      – Sparse solver in the out-of-core memory mode does lots of I/O
    – Distributed ANSYS can be highly I/O latency bound
      – Seek time to read/write each set of files causes overhead
  – Consider SSDs
    – High bandwidth and extremely low seek times
  – Consider RAID configurations
    RAID 0 – for speed
    RAID 1,5 – for redundancy
    RAID 10 – for speed and redundancy
Distributed ANSYS Performance

- Need fast hard drives to feed fast processors

- 8 million DOF
- Linear static analysis
- Sparse solver (DMP)
- Dell T5500 workstation
  12 Intel Xeon x5675 cores, 48 GB RAM, single 7.2k rpm HDD, single SSD, Win7

Distributed ANSYS Performance

- Avoid waiting for I/O to complete!
- Check to see if job is I/O bound or compute bound
  - Check output file for CPU and Elapsed times
    - When Elapsed time >> main thread CPU time
      - Consider adding more RAM or faster hard drive configuration
    - When Elapsed time ≈ main thread CPU time
      - Considering moving simulation to a machine with faster processors
      - Consider using Distributed ANSYS (DMP) instead of SMP
      - Consider running on more cores or possibly using GPU(s)

Total CPU time for main thread : 167.8 seconds
Elapsed Time (sec) = 388.000 Date = 08/21/2012
## Distributed ANSYS Performance

<table>
<thead>
<tr>
<th>Test Case</th>
<th>ANSYS 11.0</th>
<th>ANSYS 12.0</th>
<th>ANSYS 12.1</th>
<th>ANSYS 13.0 SP2</th>
<th>ANSYS 14.0</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Thermal (full model)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 MDOF</td>
<td>Time</td>
<td>4 hours</td>
<td>4 hours</td>
<td>4 hours</td>
<td>1 hour</td>
</tr>
<tr>
<td></td>
<td>Cores</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8 + 1 GPU</td>
<td>32</td>
</tr>
<tr>
<td><strong>Thermomechanical Simulation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(full model)</td>
<td>Time</td>
<td>~5.5 days</td>
<td>34.3 hours</td>
<td>12.5 hours</td>
<td>7.5 hours</td>
</tr>
<tr>
<td>7.8 MDOF</td>
<td>Iterations</td>
<td>163</td>
<td>164</td>
<td>195</td>
<td>195</td>
</tr>
<tr>
<td></td>
<td>Cores</td>
<td>8</td>
<td>20</td>
<td>64</td>
<td>128</td>
</tr>
<tr>
<td><strong>Interpolation of Boundary</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conditions**</td>
<td>Time</td>
<td>37 hours</td>
<td>37 hours</td>
<td>37 hours</td>
<td>0.2 hour</td>
</tr>
<tr>
<td></td>
<td>Load steps</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>Improved</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>algorithm</td>
<td>16</td>
</tr>
<tr>
<td><strong>Submodel: Creep-Strain</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis 5.5 MDOF</td>
<td>Time</td>
<td>~5.5 days</td>
<td>38.5 hours</td>
<td>8.5 hours</td>
<td>5.9 hours</td>
</tr>
<tr>
<td></td>
<td>Iterations</td>
<td>492</td>
<td>492</td>
<td>492</td>
<td>498</td>
</tr>
<tr>
<td></td>
<td>Cores</td>
<td>18</td>
<td>16</td>
<td>76</td>
<td>64 + 8 GPU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>128</td>
<td>256</td>
</tr>
<tr>
<td><strong>Total Time</strong></td>
<td></td>
<td>2 weeks</td>
<td>5 days</td>
<td>2 days</td>
<td>1 day</td>
</tr>
</tbody>
</table>

All runs with Sparse solver
Hardware 12.0: dual X5460 (3.16 GHz Harpertown Intel Xeon) 64GB RAM per node
Hardware 12.1 + 13.0: dual X5570 (2.93 GHz Nehalem Intel Xeon) 72GB RAM per node
ANSYS 12.0 to 14.0 runs with DDR Infiniband® interconnect
ANSYS 14.0 creep runs with NROPT,CRPL + DDOPT,METIS

Results Courtesy of MicroConsult Engineering, GmbH
Distributed ANSYS Performance

- Minimum time to solution more important than scaling

- Turbine model
- 2.1 million DOF
- Nonlinear static analysis
- 1 Loadstep, 7 substeps, 25 equilibrium iterations
- Linux cluster (8 cores per node)
Distributed ANSYS Performance

- Minimum time to solution more important than scaling

- Turbine model
- 2.1 million DOF
- Nonlinear static analysis
- 1 Loadstep, 7 substeps, 25 equilibrium iterations
- Linux cluster (8 cores per node)
GPU Accelerator Capability

• Graphics processing units (GPUs)
  – Widely used for gaming, graphics rendering
  – Recently been made available as general-purpose “accelerators”
    • Support for double precision computations
    • Performance exceeding the latest multicore CPUs

So how can ANSYS make use of this new technology to reduce the overall time to solution??
GPU Accelerator Capability

• “Accelerate” Sparse direct solver (SMP & DMP)
  – GPU is used to factor many dense “frontal” matrices
  – Decision is made automatically on when to send data to GPU
    • “Frontal matrix” too small, too much overhead, stays on CPU
    • “Frontal matrix” too large, exceeds GPU memory, only partially accelerated

• “Accelerate” PCG/JCG iterative solvers (SMP & DMP)
  – GPU is only used for sparse-matrix vector multiply (SpMV kernel)
  – Decision is made automatically on when to send data to GPU
    • Model too small, too much overhead, stays on CPU
    • Model too large, exceeds GPU memory, only partially accelerated
GPU Accelerator Capability

• **Supported hardware**
  – Currently support NVIDIA Tesla 20-series, Quadro 6000, and Quadro K5000 cards
  – Next generation NVIDIA Tesla cards (Kepler) should work with R14.5
  – Installing a GPU requires the following:
    • Larger power supply (single card needs ~250W)
    • Open 2x form factor PCIe x16 2.0 (or 3.0) slot

• **Supported platforms**
  – Windows and Linux 64-bit platforms only
    • Does not include Linux Itanium (IA-64) platform
## GPU Accelerator Capability

- **Targeted hardware**

<table>
<thead>
<tr>
<th></th>
<th>NVIDIA Tesla C2075</th>
<th>NVIDIA Tesla M2090</th>
<th>NVIDIA Quadro 6000</th>
<th>NVIDIA Quadro K5000†</th>
<th>NVIDIA Tesla K10</th>
<th>NVIDIA Tesla K20†</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Power (W)</strong></td>
<td>225</td>
<td>250</td>
<td>225</td>
<td>122</td>
<td>250</td>
<td>250</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>6 GB</td>
<td>6 GB</td>
<td>6 GB</td>
<td>4 GB</td>
<td>8 GB</td>
<td>6 to 24 GB</td>
</tr>
<tr>
<td><strong>Memory Bandwidth (GB/s)</strong></td>
<td>144</td>
<td>177.4</td>
<td>144</td>
<td>173</td>
<td>320</td>
<td>288</td>
</tr>
<tr>
<td><strong>Peak Speed SP/DP (GFlops)</strong></td>
<td>1030/515</td>
<td>1331/665</td>
<td>1030/515</td>
<td>2290/95</td>
<td>4577/190</td>
<td>5184/1728</td>
</tr>
</tbody>
</table>

† These NVIDIA “Kepler” based products are not released yet, so specifications may be incorrect.
GPU Accelerator Capability

- GPUs can offer significantly faster time to solution

- 6.5 million DOF
- Linear static analysis
- Sparse solver (DMP)
- 2 Intel Xeon E5-2670 (2.6 GHz, 16 cores total), 128 GB RAM, SSD, 4 Tesla C2075, Win7

Relative Speedup

<table>
<thead>
<tr>
<th></th>
<th>2 cores (no GPU)</th>
<th>8 cores (no GPU)</th>
<th>8 cores (1 GPU)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.5</td>
<td>2.6x</td>
<td>3.8x</td>
</tr>
</tbody>
</table>
GPU Accelerator Capability

- GPUs can offer significantly faster time to solution

- 11.8 million DOF
- Linear static analysis
- PCG solver (DMP)
- 2 Intel Xeon E5-2670 (2.6 GHz, 16 cores total), 128 GB RAM, SSD, 4 Tesla C2075, Win7

**GPU Performance**

- Relative Speedup
- 2 cores (no GPU)
- 8 cores (1 GPU)
- 16 cores (4 GPUs)

- 2.7x
- 5.2x
GPU Accelerator Capability

• Supports majority of ANSYS users
  – Covers both sparse direct and PCG iterative solvers
  – Only a few minor limitations

• Ease of use
  – Requires at least one supported GPU card to be installed
  – Requires at least one HPC pack license
  – No rebuild, no additional installation steps

• Performance
  – ~10-25% reduction in time to solution when using 8 CPU cores
  – *Should never slow down your simulation!*
Design Optimization

• How will you use all of this computing power?

Higher fidelity  Full assemblies  More nonlinear

Design Optimization Studies
HPC Licensing

- ANSYS HPC Packs enable high-fidelity insight
  - Each simulation consumes one or more packs
  - Parallel enabled increases quickly with added packs
- Single solution for all physics and any level of fidelity
- Flexibility as your HPC resources grow
  - Reallocate Packs, as resources allow
HPC Parametric Pack Licensing

- Scalable, like ANSYS HPC Packs
  - Enhances the customer’s ability to include many design points as part of a single study
  - Ensure sound product decision making
- Amplifies complete workflow
  - Design points can include execution of multiple products (pre, solve, HPC, post)
  - Packaged to encourage adoption of the path to robust design!
HPC Revolution

The right combination of algorithms and hardware leads to maximum efficiency
HPC Revolution

*Every computer today is a parallel computer*

*Every simulation in ANSYS can benefit from parallel processing*