Early Experiences with Trinity - The First Advanced Technology Platform for the ASC Program


Sandia National Laboratories

CUG 2016, London, UK
Trinity

- Cray XC40
- Total of about 19000 nodes
  - About half are Intel Haswell with 2 processors per node and 16 cores per processor running at 2.3 GHz and 128 GB memory per node
  - About half are 60+ core Intel Knights Landing processors - will be delivered later this year
  - Cray Aries Dragonfly interconnect
- Expect greater than 30 PetaFlops peak
Cielo and Chama

- Cielo is a Cray XE6 and is our current generation capability machine
  - 8894 nodes with 2 oct-core AMD Magny-Cours processors running at 2.4 GHz
  - Cray Gemini 3D torus interconnect
- Chama is current generation capacity machine
  - Tri-Lab Capacity Cluster
  - 1232 nodes with 2 Intel Sandy Bridge 8 core processors running at 2.6 GHz
  - Qlogic QDR-InfiniBand Fat-Tree interconnect
# Comparison of Machines

<table>
<thead>
<tr>
<th>System</th>
<th>Cielo</th>
<th>Chama</th>
<th>Trinity (Phase-I)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Nodes</td>
<td>8,894</td>
<td>1,232</td>
<td>9,408</td>
</tr>
<tr>
<td>Total Cores</td>
<td>142,304</td>
<td>19,712</td>
<td>301,056</td>
</tr>
<tr>
<td>Processor</td>
<td>AMD Magney-Cours</td>
<td>Intel Sandy Bridge</td>
<td>Intel Haswell</td>
</tr>
<tr>
<td>Processor ISA</td>
<td>SSE4a</td>
<td>AVX</td>
<td>AVX-2</td>
</tr>
<tr>
<td>Clock Speed (GHz)</td>
<td>2.40</td>
<td>2.60</td>
<td>2.30</td>
</tr>
<tr>
<td>Cores/Socket</td>
<td>8 (2x4)</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Cores/Node</td>
<td>16</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>Peak Node (GFLOPs)</td>
<td>153.6</td>
<td>332.8</td>
<td>1,177.6</td>
</tr>
<tr>
<td>Memory</td>
<td>DR3-1333</td>
<td>DDR3-1600</td>
<td>DDR4-2133</td>
</tr>
<tr>
<td>Channels/Socket</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Interconnect</td>
<td>Cray Gemini</td>
<td>Qlogic QDR-InfiniBand</td>
<td>Cray Aries</td>
</tr>
<tr>
<td>Topology</td>
<td>3D-Torus</td>
<td>Fat-Tree</td>
<td>DragonFly</td>
</tr>
</tbody>
</table>
Comments on STREAMS

- Peak compute speed for a Trinity node is 7.7 times faster than a node of Cielo and 3.54 times faster than a node of Chama.

- STREAM results per node on Trinity are >2X that of Cielo and more than 1.5X that of Chama.

- STREAM results per core on Trinity are about 11% higher than a core of Cielo and about 22% lower than a core of Chama.
Comments on MPI Benchmarks

- Chama shows better Ping Pong Latency than Cielo and Trinity for small messages, but for large messages, Trinity has better latency.

- Likewise, Chama has better Ping Pong Bandwidth for small messages and Trinity has more than twice the Bandwidth for large messages.

- Allreduce operations are 2 to 10 times faster on 256 ranks of Trinity.
Focus Codes

- Focus on Production SIERRA applications
  - SIERRA/Solid Mechanics (SM)
  - SIERRA/Aerodynamics
  - SIERRA/Structural Dynamics (SD)

- SIERRA is a large C++ framework
  - provides framework for several codes
  - Includes several Third Party Libraries
  - Contains common C++ classes and methods
  - Common infrastructure for parallel codes
SIERRA/SM (Solid Mechanics)

- A general purpose massively parallel nonlinear solid mechanics finite element code for explicit transient dynamics, implicit transient dynamics and quasi-statics analysis.
- Built upon extensive material, element, contact and solver libraries for analyzing challenging nonlinear mechanics problems for normal, abnormal, and hostile environments.
- Similar to LSDyna or Abaqus commercial software systems.
Summary of Sierra/Aero

• Unstructured meshes
• One and two equation turbulence models
• LES and Hybrid RANS
• Uses either FETI or Trilinos for sparse matrix operations and solvers.
• Assembly is substantial portion of the computational cost.

Turbulent flow past a cavity
SIERRA/SD Domain Areas

- General Structural Dynamics, Finite Elements
  - Vibrations, normal modes, implicitly integrated transient dynamics, frequency response analysis
  - Shells, Solids, Beams, Point Masses
  - Complicated Large Structures
  - Typically many constraint equations
- Acoustics and Structural Acoustics
  - Even larger systems
  - More constraints
  - Infinite Elements (nonsymmetric)
- Optimization, UQ and Inverse Methods
  - Adjoint methods
  - Material and Parameter inversion
  - Verification and Validation
Code Characteristics

- SIERRA/SM extensively uses sparse direct solvers
  - the iterative solve requires a local solve, coarse solve, and a preconditioner
  - The preconditioning step dominates the cost (>90%).
- SIERRA/Aero uses Trilinos solvers
  - GMRES for solver with Symmetric Gauss-Seidel preconditioner
- SIERRA/SD uses Domain Decomposition solver for eigensolve
  - About 84% of non-MPI time spent in solver which makes use of BLAS routines
<table>
<thead>
<tr>
<th>Code/problem</th>
<th>Cores</th>
<th>Cielo</th>
<th>Chama</th>
<th>Trinity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aero implicit</td>
<td>128</td>
<td>1834.0</td>
<td>961.2</td>
<td>874.4</td>
</tr>
<tr>
<td>Aero explicit</td>
<td>128</td>
<td>527.0</td>
<td>294.6</td>
<td>278.2</td>
</tr>
</tbody>
</table>

- Both Chama and Trinity are about twice as fast as Cielo
- For explicit problem, compute time is similar, but MPI time is difference between Chama and Trinity
- For implicit problem, Trinity is 10% faster than Chama for compute and 14% to 20% faster for MPI time
Aero Times (in seconds)

<table>
<thead>
<tr>
<th>Code/problem</th>
<th>Cores</th>
<th>Cielo</th>
<th>Chama</th>
<th>Trinity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aero hex</td>
<td>512</td>
<td>658.2</td>
<td>355.9</td>
<td>351.9</td>
</tr>
<tr>
<td>Aero mixed</td>
<td>512</td>
<td>390.7</td>
<td>213.3</td>
<td>192.4</td>
</tr>
</tbody>
</table>

- Both Chama and Trinity are about twice as fast as Cielo.
- For both problems, compute time is similar on Chama and Trinity and MPI time is about 20% less on Trinity than on Chama.
- MPI time is about 20% for the hex problem and 55% for the mixed.
### SM Times (in seconds)

<table>
<thead>
<tr>
<th>Code/problem</th>
<th>Cores</th>
<th>Cielo</th>
<th>Chama</th>
<th>Trinity</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM</td>
<td>16</td>
<td>1118.3</td>
<td>657.5</td>
<td>598.4</td>
</tr>
<tr>
<td>SM refined</td>
<td>128</td>
<td>2332.1</td>
<td>1452.1</td>
<td>1369.1</td>
</tr>
</tbody>
</table>

- Both Chama and Trinity are about 1.7 times as fast as Cielo
- For refined problem, MPI takes about 55% of the time
- On Trinity and Chama, the compute time is similar, but the MPI time is larger on Chama than Trinity
  - Lots of small to medium sized messages
SD Times (in seconds)

<table>
<thead>
<tr>
<th>Code/problem</th>
<th>Cores</th>
<th>Cielo</th>
<th>Chama</th>
<th>Trinity</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD</td>
<td>120</td>
<td>993.0</td>
<td>451.0</td>
<td>540.0</td>
</tr>
</tbody>
</table>

- Both Chama and Trinity are about twice as fast as Cielo.
- MPI and compute times are larger on Trinity than Chama, but the MPI time includes wait times, which is an indication of load imbalance.
- Most of compute time is spent in a solver which has a large number of DGEMM calls.
Comments on Performance

- All of the codes show Trinity faster than Cielo by a factor of about 2 on half as many nodes
  - Effectively a factor of four per node

- Most of the codes show Trinity 5% to 12% faster than Chama on half as many nodes
  - About a factor of two per node

- Sierra/SD about 18% slower on Trinity than on Chama
  - Slightly worse than the difference in clock speeds
Summary

- We ported three production codes to our new capability machine and compared its performance to our current machines
  - We got about a factor of four performance improvement per node over our current capability machine and a factor of two over our current capacity machine

- We are investigating the performance of SIERRA/SD

- We have started working on the Knight’s Landing portion of the machine
Exceptional service in the national interest