Experiences with Sandia National Laboratories HPC applications and MPI performance

Mahesh Rajan, Doug Doerfler, Richard Barrett, Joel Stevenson, Anthony Agelastos, Ryan Shaw and Hal Meyer

MVAPICH User Group Meeting, Aug 25-27, 2014, Columbus OH
Sandia’s History

exceptional service in the national interest.
Sandia’s Sites

Albuquerque, New Mexico

Waste Isolation Pilot Plant, Carlsbad, New Mexico

Livermore, California

Tonopah, Nevada

Pantex, Texas

9/3/2014

Sandia Unclassified Unlimited Release
Research Disciplines Drive Capabilities

- Computer Science
- Materials
- Engineering Sciences
- Micro Electronics
- Bioscience
- Pulsed Power

High Performance Computing
Nanotechnologies & Microsystems
Extreme Environments

9/3/2014
Sandia Unclassified Unlimited Release
# Capacity Computing Resources (6-2014)

<table>
<thead>
<tr>
<th>System</th>
<th>Vendor</th>
<th>Nodes</th>
<th>Processor cores total</th>
<th>Processor:Sockets:Cores/Socket &amp; Interconnect</th>
<th>Memory/Node</th>
<th>Memory/Core</th>
<th>TFLOPS</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Sky</td>
<td>Sun</td>
<td>2,823</td>
<td>22,584</td>
<td>2.93 GHz Intel Nehalem:2S:4C; Mellanox ConnectX IB QDR</td>
<td>12</td>
<td>1.5</td>
<td>264</td>
<td>Apr-10</td>
</tr>
<tr>
<td>Glory</td>
<td>Appro</td>
<td>272</td>
<td>4,352</td>
<td>2.2 GHz AMD:4S:4C</td>
<td>32</td>
<td>2</td>
<td>38</td>
<td>Jan-09</td>
</tr>
<tr>
<td>Chama</td>
<td>Appro</td>
<td>1,232</td>
<td>19,712</td>
<td>2.6 GHz Intel Sandy Bridge:2S:8C; Qlogic 4X IB QDR</td>
<td>32</td>
<td>2</td>
<td>392</td>
<td>Sep-12</td>
</tr>
<tr>
<td>Uno</td>
<td>Dell</td>
<td>251</td>
<td>3,344</td>
<td>2.7 GHz Intel Sandy Bridge:2S:8C/4S:8C</td>
<td>64/128</td>
<td>4/8</td>
<td>71</td>
<td>TBD</td>
</tr>
</tbody>
</table>

**Black Total:** 4,578 49,992 765

<table>
<thead>
<tr>
<th>System</th>
<th>Vendor</th>
<th>Nodes</th>
<th>Processor cores total</th>
<th>Processor:Sockets:Cores/Socket &amp; Interconnect</th>
<th>Memory/Node</th>
<th>Memory/Core</th>
<th>TFLOPS</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Sky</td>
<td>Sun</td>
<td>519</td>
<td>4,152</td>
<td>2.93 GHz Intel Nehalem:2S:4C</td>
<td>12</td>
<td>1.5</td>
<td>48</td>
<td>Nov-10</td>
</tr>
<tr>
<td>Jemez</td>
<td>HP</td>
<td>288</td>
<td>4608</td>
<td>2.6 GHz Intel Sandy Bridge:2S:8C; Mellanox ConnectX-3 IB FDR</td>
<td>32</td>
<td>2</td>
<td>95</td>
<td>Jun-14</td>
</tr>
<tr>
<td>Unity</td>
<td>Appro</td>
<td>272</td>
<td>4,352</td>
<td>2.2 GHz AMD:4S:4C</td>
<td>32</td>
<td>2</td>
<td>38</td>
<td>Mar-09</td>
</tr>
<tr>
<td>Whitney</td>
<td>Appro</td>
<td>272</td>
<td>4,352</td>
<td>2.2 GHz AMD:4S:4C</td>
<td>32</td>
<td>2</td>
<td>38</td>
<td>Mar-09</td>
</tr>
<tr>
<td>Pecos</td>
<td>Appro</td>
<td>1,232</td>
<td>19,712</td>
<td>2.6 GHz Intel Sandy Bridge:2S:8C</td>
<td>32</td>
<td>2</td>
<td>392</td>
<td>Sep-12</td>
</tr>
</tbody>
</table>

**Red Total:** 2,583 37,176 611

<table>
<thead>
<tr>
<th>System</th>
<th>Vendor</th>
<th>Nodes</th>
<th>Processor cores total</th>
<th>Processor:Sockets:Cores/Socket &amp; Interconnect</th>
<th>Memory/Node</th>
<th>Memory/Core</th>
<th>TFLOPS</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Mesa</td>
<td>Sun</td>
<td>1,920</td>
<td>15,360</td>
<td>2.93 GHz Intel Nehalem:2S:4C</td>
<td>12</td>
<td>1.5</td>
<td>180</td>
<td>May-10</td>
</tr>
</tbody>
</table>

**Green Total:** 1,920 15,360 180

<table>
<thead>
<tr>
<th>System</th>
<th>Vendor</th>
<th>Nodes</th>
<th>Processor cores total</th>
<th>Processor:Sockets:Cores/Socket &amp; Interconnect</th>
<th>Memory/Node</th>
<th>Memory/Core</th>
<th>TFLOPS</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gila</td>
<td>HP</td>
<td>48</td>
<td>1,152</td>
<td>2.3 GHz AMD 6176:2S:12C</td>
<td>24</td>
<td>2</td>
<td>11</td>
<td>Apr-13</td>
</tr>
<tr>
<td>Dark Bridge</td>
<td>Appro</td>
<td>924</td>
<td>14,784</td>
<td>2.6 GHz Intel Sandy Bridge:2S:8C</td>
<td>64</td>
<td>4</td>
<td>294</td>
<td>Nov-12</td>
</tr>
<tr>
<td>Dark Sand</td>
<td>Appro</td>
<td>924</td>
<td>14,784</td>
<td>2.6 GHz Intel Sandy Bridge:2S:8C</td>
<td>64</td>
<td>4</td>
<td>294</td>
<td>Dec-13</td>
</tr>
</tbody>
</table>

**Orange Total:** 1,896 30,720 599

**TOTAL:** 10,977 133,248 2,155
Sandia/LANL ACES Capability System: Cray XE6

- **Topology**
  - Gemini High-Speed Interconnect
  - Phase 2: 24x16(8)x24 3D Torus

- **Node: dual-socket, AMD Magny-Cours**
  - 16 total cores (8 per socket)
  - 2.4 GHz core clock rate
  - 8 channels (4 per socket)
    - 1333 MHz DDR3 memory
  - 4 FLOPS per clock per core
  - 32 GB total memory
  - 153.6 GF peak FP
  - 85.3 GB/s peak memory BW

<table>
<thead>
<tr>
<th>Total # of racks</th>
<th>96</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total # of compute blades</td>
<td>2,235</td>
</tr>
<tr>
<td># of compute nodes</td>
<td>8,940</td>
</tr>
<tr>
<td># of cores</td>
<td>143,040</td>
</tr>
<tr>
<td>Hostname /Integrator</td>
<td>Num Nodes</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------</td>
</tr>
<tr>
<td><strong>Volta / Cray XC30m</strong></td>
<td>56</td>
</tr>
<tr>
<td><strong>Teller /Penguin</strong></td>
<td>104</td>
</tr>
<tr>
<td><strong>Shannon /Appro</strong></td>
<td>32</td>
</tr>
<tr>
<td><strong>Compton / appro</strong></td>
<td>42</td>
</tr>
<tr>
<td><strong>Curie /Cray XK7</strong></td>
<td>52</td>
</tr>
</tbody>
</table>
Sandia’s Advanced Architecture Systems and Mantevo mini-apps

• Project Aim: Study Future Architectures, Power and Programming Models
  – Study low power cores based nodes
  – AMD APU, Intel Xeon Phi, NVIDIA GPU, ...
  – Lighter-weight hosts, e.g. ARM?; hostless?
  – Multi-level memory hierarchies
  – Interconnects
  – Programming models:
    – Memory model abstractions, e.g. Kokkos
    – Task parallel, over-decomposition of smaller granularity data parallel computation.
## Miniapps: Tools enabling exploration

<table>
<thead>
<tr>
<th>Focus</th>
<th>Proxy for a key app performance issue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intent</td>
<td>Tool for codesign: output is information</td>
</tr>
<tr>
<td>Scope of change</td>
<td>Any and all</td>
</tr>
<tr>
<td>Size</td>
<td>A few thousand lines of code</td>
</tr>
<tr>
<td>Availability</td>
<td>Open source (LGPL)</td>
</tr>
<tr>
<td>Developer/owner</td>
<td>Application team</td>
</tr>
<tr>
<td>Life span</td>
<td>Until its no longer useful</td>
</tr>
</tbody>
</table>

**Related:**

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Output: metric to be ranked.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compact app</td>
<td>Application relevant answer.</td>
</tr>
<tr>
<td>Skeleton app</td>
<td>Inter-process comm, application “fake”</td>
</tr>
<tr>
<td>Proxy app</td>
<td>Über notion</td>
</tr>
</tbody>
</table>
## Mantevo 2.0+ status; http://mantevo.org

<table>
<thead>
<tr>
<th>miniApp or miniDriver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cleverleaf 1.0</td>
<td>Eulerian on structured grid with AMR</td>
</tr>
<tr>
<td>CloverLeaf 1.0.1</td>
<td>Compressible Euler eqns, explicit 2\textsuperscript{nd} order accurate</td>
</tr>
<tr>
<td>CoMD 1.1</td>
<td>Molecular dynamics (SPaSM)</td>
</tr>
<tr>
<td>EpetraBenchmarkTest 1.0</td>
<td>Exercises Epetra sparse and dense kernels.</td>
</tr>
<tr>
<td>HPCCG 1.0</td>
<td>Unstructured implicit finite element</td>
</tr>
<tr>
<td>miniFE 2.0</td>
<td>Implicit finite element solver</td>
</tr>
<tr>
<td>miniGhost 1.0.1</td>
<td>FDM/FVM explicit (halo exchange focus)</td>
</tr>
<tr>
<td>miniMD 1.2</td>
<td>Molecular dynamics (Lennard-Jones)</td>
</tr>
<tr>
<td>miniXyce 1.0</td>
<td>SPICE-style circuit simulator</td>
</tr>
<tr>
<td>miniAMR</td>
<td>Adaptive mesh refinement of an Eulerian mesh</td>
</tr>
<tr>
<td>miniSMAC</td>
<td>FD 2D incompressible N/S on a structured grid.</td>
</tr>
<tr>
<td>PathFinder</td>
<td>Signature search</td>
</tr>
<tr>
<td>miniAero</td>
<td>3D unstructured FV R-K 4\textsuperscript{th} order time marching, inviscid Roe Flux</td>
</tr>
<tr>
<td>miniContact</td>
<td>Solid mechanics</td>
</tr>
<tr>
<td>miniExDyn-FE</td>
<td>Explicit Dynamics (Kokkos-based)</td>
</tr>
<tr>
<td>miniITC-FE</td>
<td>Implicit Thermal Conduction (Kokkos-based)</td>
</tr>
<tr>
<td>phdMesh</td>
<td>Explicit FEM: contact detection</td>
</tr>
</tbody>
</table>

2.0 release

Coming soon: miniSAT, miniTri
MPI Challenges with SNL HPC Systems and applications

• Limited staff supporting a variety of systems with different application requirements
  – Example: Chama, 1232 Node Capacity Cluster (software managed through modules)
    • MPI: OpenMPI(1.4,1.5,1.6), MVAPICH(1.2,1.7)
    • Compilers: GNU, Intel, PGI
    • Tools: mpiP, Vtune, HPCToolkit, OpenSpeedShop, TAU, MAP

• Makes the job of optimal resource utilization a challenge with so many possible combinations of compilers, libraries, tools and applications running on our many systems
Use of MVAPICH at SNL on Chama

- Used MVAPICH psm/1.2 when we acquired and installed the TLCC2 systems in 2012/13
  - For all initial evaluation of the new system
  - For Top500 Linpack measurements: Feb 24; 332 TFLOPS; 81.11 % Efficiency on 1230 Nodes; Chama #31 on the top500 (Feb 2012)
  - For early investigations of throughput improvement with doubling the memory per core

- FOR SNL’s Neutron Generator code; Aleph
  - Memory issues when tried with OpenMPI
  - Also later discovered an internal bug with an OpenMPI sub communicator

- Early scaling studies on Chama with MVAPICH using Cielo acceptance applications showed excellent scaling. First time such good scaling seen with commodity clusters. Below scaling plots for a few Cielo acceptance apps (See SC2013 PMBS paper by Rajan, et.al.)
A FEW APPLICATIONS TO ILLUSTRATE MPI CHALLENGES SEEN
SIERRA/Aria- Implicit CFD; Key to performance - 1k size MPI message rate; At max scale Chama is 2.6X better than Cielo; MPI message rate all important for solver based codes

![SIERRA/Aria Strong Scaling](image_url)

![osu_mbw_mr; message rate ratio](image_url)
Sandia implicit codes benefit from MVAPICH
potential 1.5X gain in message rate at the 1k to 4k message size of interest
Intent is not to compare MPI libraries but find ‘tunables’ that can benefit users
used: mvapich2-intel-ofa/1.7 and openmpi-intel/1.6

osu_mbw_mr MVAPICH/OpenMPI ratio; Jemez
Another implicit CFD code showing benefit of MVAPICH
Sierra NALU: Low Mach CFD

Simulation description & parameters:
• Jet-in-crossflow simulation counterpart to Su & Mungal experimental results
• GMRES w/ML, Gram-Schmidt preconditioners
• Mesh: 152,463,520 hex-8 unstructured elem.
• 1,536 MPI ranks (1 rank per core) on Chama
• Each ensemble was run alternating MPI within the same node allocation, e.g., MVAPICH, OpenMPI, MVAPICH, OpenMPI,...

• Simulation observations:
  – On average, MVAPICH is 2-4% faster than OpenMPI
  – MVAPICH shows lower run-time variability
    • Variability with simulation runtimes impedes many things including performance regression testing
    • When the outliers in the run-time data are omitted MVAPICH’s standard deviation is 2x-3x less than OpenMPI,
    • mpiP profile showed: ~60% of run time in MPI; 82% of MPI time in global operations (mostly MPI_Allreduce)
  – Asynchronous MPI global operations in MPI 3.0 could have a big impact
    • Potential reduction in variability due to OS noise/jitter
    • Asynchronous global operations may help reduce total time in Allreduce through better overlap of computations & communications
A most demanding application from an MPI perspective: Sierra/Adagio; Contact Algorithm

- ECSL Model
  - 2,158,543 elements Hex elements,
  - bolts, springs Contact,
  - multiple materials,
  - many parts, preload
- Sierra::Newton::Apst_Contact::contact_search takes 93% of run time out of which 76% is MPI;
- Contact search MPI calls are the dominant overhead.
- MPI_Allreduce time including 'sync time' due to load imbalance is the main scaling inhibitor. See chart below
- Efficient Asynchronous MPI collectives are needed!!

**Adagio Speed-Up on Chama;**

ECSL model; 2,158,543 Elements

---

**Adagio ECSL Model MPI Profile**

- Allreduce
- pt2pt
- Wait
- Barrier
- Compute

---

**Percentage of run time in MPI**

- 1024
- 512
- 256
- 128
- 64
- 32

9/3/2014 Sandia Unclassified Unlimited Release
SNL Open Source CFD code NALU (openJet Problem) on Sequoia and Cielo; weak scaling. Edge Creation uses stk mesh; usage of MPI_Alltoallv Scaling inhibitor on Cielo Alltoallv; Sequoia is 5X @ 64PEs to 40X @16kPEs faster

Nalu Edge Creation Weak Scaling Parallel Efficiency

Cielo and Sequoia MPI 8 Bytes Alltoallv Performance
A typical case of interest with a grid of dimension of 1024x1024x1024; Forward and backward transform on 512 CORES (16x32 processor grid) on Jemez; **MVAPICH was about 20% slower than OpenMPI.** Alltoallv message exchange sizes are 32K Bytes. Would like to understand sudden poorer performance with MVAPICH at 16K+ message size.

Are there ENV settings that would alleviate this problem?

![Graph showing Jemez: Alltoallv Run Time Ratio: MVAPICH/OpenMPI](image)
CTH: MVAPICH gives 2% to 11% improvement over OpenMPI

- CTH is a multi-material, large deformation, strong shock wave, solid mechanics code developed at Sandia National Laboratories.
- The code is explicit and uses finite difference (volume) for the numerical simulation of the high-rate response of materials to impulsive loads.
- CTH modeling capabilities include shock waves, low mach flows, multi-phase materials, mixed-phase materials, elastic-plastic solids, visco-plastic solids, visco-elastic solids, fracture, failure, high explosive detonation and initiation in 1D, 2D, and 3D geometries.
- The code is widely used in the DOE and DOD communities in the development of explosives, blast and fragmenting warheads, kinetic energy penetrators, vehicle armor systems, and protective structures.
- Communication patterns are problem dependent, although in general, processors exchange information with up to six other processors in the domain and messaging is dominated by large messages (several MB in size), with some small message Allreduce at scale.

![Graph showing improvement over OpenMPI](image)

Two to three runs each of fireball and fragmented pipe were performed at three core counts (256, 512, 1024) for both OpenMPI and MVAPICH. The scenarios were designed so that run times ranged between 1 hour and 8 hours. The minimum run times for OpenMPI and MVAPICH were then compared.

The “fireball” CTH simulation performed in this investigation is a 3D shock physics problem with AMR. A computational model was used to calculate the three-dimensional evolution of a nuclear fireball (shown in red, bottom left). CTH version 10.3p (Feb. 2013) was used for this calculation. This particular CTH problem was chosen for the investigation because it is representative of a large class of shock physics problems that run regularly on Sandia production platforms. This problem is a production simulation capable of running at a continuum of core counts. We used 256, 512, and 1024 cores for this investigation.

The “fragmented pipe” CTH simulation performed in this investigation is a 3D shock physics problem with AMR. A computational model was used to calculate the three-dimensional evolution of a fragmented pipe blowing apart in firecracker-like fashion (shown bottom right). CTH version 10.3p (Feb. 2013) was used for this calculation. This particular CTH problem was chosen for the investigation because it is representative of a large class of shock physics problems that run regularly on Sandia production platforms. We used 256, 512, and 1024 cores for this investigation.
Allreduce used quite heavily with Sandia apps; potential Allreduce time reduction for 128 to 4k PEs.

Intent is not to compare MPI libraries but find ‘tunables’ that can benefit users

used: mvapich2-intel-ofa/1.7 and openmpi-intel/1.6

8 Bytes Allreduce time ratio OpenMPI/MVAPICH

![Graph showing time ratio vs. MPI Tasks]

Allreduce time ratio at 4K PEs; OpenMPI/MVAPICH

![Graph showing time ratio vs. Message Size, Bytes]
Conclusions

• Our Goal: Use of MPI library that is robust and performs well; help users with optimizing performance for production runs on all our HPC systems

• Shared a few application use cases that stress MPI performance; e.g. collectives and 1k-4k messages

• Some benchmark data collected point to potential benefit for application performance with MVAPICH

• Would like from MVAPICH group & from MUG meeting
  – Suggestions on how users can extract best performance
  – Optimal environment variable settings
  – MPI 3.0 release and potential benefits
  – Collaboration with Sandia on Advanced architectures and use of Mantevo mini-apps