High Performance Computing and Data Resources at SDSC

Mahidhar Tatineni (mahidhar@sdsc.edu)

SDSC Summer Institute

August 05, 2013
HPC Resources at SDSC
Hardware Overview

HPC Systems: Gordon, Trestles
Parallel Filesystems: Data Oasis
Gordon – An Innovative Data Intensive Supercomputer

- Designed to accelerate access to massive amounts of data in areas of genomics, earth science, engineering, medicine, and others
- Emphasizes memory and IO over FLOPS.
- Appro integrated 1,024 node Sandy Bridge cluster
- 300 TB of high performance Intel flash
- Large memory supernodes via vSMP Foundation from ScaleMP
- 3D torus interconnect from Mellanox
- In production operation since February 2012
- Funded by the NSF and available through the NSF Extreme Science and Engineering Discovery Environment program (XSEDE)
## Gordon System Specification

### INTEL SANDY BRIDGE COMPUTE NODE

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sockets</td>
<td>2</td>
</tr>
<tr>
<td>Cores</td>
<td>16</td>
</tr>
<tr>
<td>Clock speed</td>
<td>2.6</td>
</tr>
<tr>
<td>DIMM slots per socket</td>
<td>4</td>
</tr>
<tr>
<td>DRAM capacity</td>
<td>64 GB</td>
</tr>
</tbody>
</table>

### INTEL FLASH I/O NODE

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAND flash SSD drives</td>
<td>16</td>
</tr>
<tr>
<td>SSD capacity per drive</td>
<td>300 GB</td>
</tr>
<tr>
<td>Capacity per node</td>
<td>4.8 TB</td>
</tr>
<tr>
<td>Total</td>
<td>300 TB</td>
</tr>
<tr>
<td>Flash bandwidth per drive</td>
<td>270 MB/s</td>
</tr>
<tr>
<td>(read/write)</td>
<td>210 MB/s</td>
</tr>
<tr>
<td>Flash bandwidth per node</td>
<td>2.7 / 3.3 GB/s</td>
</tr>
</tbody>
</table>

### SMP SUPER-NODE

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute nodes</td>
<td>32</td>
</tr>
<tr>
<td>I/O nodes</td>
<td>2</td>
</tr>
<tr>
<td>Addressable DRAM</td>
<td>2 TB</td>
</tr>
<tr>
<td>Addressable memory including flash</td>
<td>12 TB</td>
</tr>
</tbody>
</table>

### GORDON

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute Nodes</td>
<td>1,024</td>
</tr>
<tr>
<td>Total compute cores</td>
<td>16,384</td>
</tr>
<tr>
<td>Peak performance</td>
<td>341 TF</td>
</tr>
<tr>
<td>Aggregate memory</td>
<td>64 TB</td>
</tr>
</tbody>
</table>

### INFINIBAND INTERCONNECT

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregate torus BW</td>
<td>9.2 TB/s</td>
</tr>
<tr>
<td>Type</td>
<td>Dual-Rail QDR InfiniBand</td>
</tr>
<tr>
<td>Link Bandwidth</td>
<td>8 GB/s (bidirectional)</td>
</tr>
<tr>
<td>Latency (min-max)</td>
<td>1.25 µs – 2.5 µs</td>
</tr>
</tbody>
</table>

### DISK I/O SUBSYSTEM

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total storage</td>
<td>4.5 PB (raw)</td>
</tr>
<tr>
<td>I/O bandwidth</td>
<td>100 GB/s</td>
</tr>
<tr>
<td>File system</td>
<td>Lustre</td>
</tr>
</tbody>
</table>
The Memory Hierarchy of a Typical Supercomputer

- Shared memory Programming (single node)
- Message passing programming
- Disk I/O
The Memory Hierarchy of Gordon

- Registers (1 cycle)
- Caches (2-10 cycles)
- Memory (100 cycles)
- Remote Memory (10,000 cycles)
- Flash Drives (100,000 cycles)
- Spinning Disks (10,000,000 cycles)

BIG DATA

Shared memory Programming (multiple nodes)

Disk I/O
Gordon Design Highlights

- 1,024 2S Xeon E5 (Sandy Bridge) nodes
- 16 cores, 64 GB/node
- Intel Jefferson Pass mobo
- PCI Gen3

- 3D Torus
- Dual rail QDR

- Large Memory vSMP Supernodes
- 2TB DRAM
- 10 TB Flash

- 300 GB Intel 710 eMLC SSDs
- 300 TB aggregate

- 64, 2S Westmere I/O nodes
- 12 core, 48 GB/node
- 4 LSI controllers
- 16 SSDs
- Dual 10GbE
- SuperMicro mobo
- PCI Gen2

“Data Oasis”
Lustre PFS
100 GB/sec, 4 PB

2013 Summer Institute: Discover Big Data, August 5-9, San Diego, California
Gordon Architecture: 3D Torus of Switches

- Linearly expandable
- Simple wiring pattern
- Short Cables- Fiber Optic cables generally not required
- Lower Cost : 40% as many switches, 25% to 50% fewer cables
- Works well for localized communication
- Fault Tolerant within the mesh with 2QoS Alternate Routing
- Fault Tolerant with Dual-Rails for all routing algorithms

3rd dimension wrap-around not shown for clarity
Gordon vSMP Supernode

vSMP aggregation SW

ION

Dual WM IOP
4.8 TB flash SSD

ION

Dual WM IOP
4.8 TB flash SSD
## Trestles - System Description

<table>
<thead>
<tr>
<th>System Component</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMD MAGNY-COURS COMPUTE NODE</td>
<td></td>
</tr>
<tr>
<td>Sockets</td>
<td>4</td>
</tr>
<tr>
<td>Cores</td>
<td>32</td>
</tr>
<tr>
<td>Clock Speed</td>
<td>2.4 GHz</td>
</tr>
<tr>
<td>Flop Speed</td>
<td>307 Gflop/s</td>
</tr>
<tr>
<td>Memory capacity</td>
<td>64 GB</td>
</tr>
<tr>
<td>Memory bandwidth</td>
<td>171 GB/s</td>
</tr>
<tr>
<td>STREAM Triad bandwidth</td>
<td>100 GB/s</td>
</tr>
<tr>
<td>Flash memory (SSD)</td>
<td>120 GB</td>
</tr>
</tbody>
</table>

### FULL SYSTEM

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Total compute nodes</td>
<td>324</td>
</tr>
<tr>
<td>Total compute cores</td>
<td>10,368</td>
</tr>
<tr>
<td>Peak performance</td>
<td>100 Tflop/s</td>
</tr>
<tr>
<td>Total memory</td>
<td>20.7 TB</td>
</tr>
<tr>
<td>Total memory bandwidth</td>
<td>55.4 TB/s</td>
</tr>
<tr>
<td>Total flash memory</td>
<td>39 TB</td>
</tr>
</tbody>
</table>

### QDR INFINIBAND INTERCONNECT

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Topology</td>
<td>Fat tree</td>
</tr>
<tr>
<td>Link bandwidth</td>
<td>8 GB/s (bidirectional)</td>
</tr>
<tr>
<td>Peak bisection bandwidth</td>
<td>5.2 TB/s (bidirectional)</td>
</tr>
<tr>
<td>MPI latency</td>
<td>1.3 us</td>
</tr>
</tbody>
</table>

### DISK I/O SUBSYSTEM (SDSC’s Data Oasis)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>File systems</td>
<td>NFS, Lustre</td>
</tr>
<tr>
<td>Storage capacity (usable)</td>
<td>150 TB: Dec 2010, 2PB : June 2011</td>
</tr>
<tr>
<td>I/O bandwidth</td>
<td>50 GB/s (June 2011)</td>
</tr>
</tbody>
</table>
Trestles - Configuring for productivity

- Limit allocation percentage (of the theoretically available SUs)
  - Common practice of allocating ~80% (or more) can result in high wait times.
  - We monitor wait times via expansion factors and cap the total time allocated. This allows the scheduler to better satisfy a mix of scheduling modes.

- Limit (non-gateway) projects to 1.5M SUs/year (~2.5% total)
  - Projects w/ large % allocations dominate resource consumption and drive up queue waits
  - This cap creates room for more users & helps reduce queue waits across the board.

- Configure the job queues and resource schedulers for lower expansion factors and generally faster turnaround.
  - System data and user feedback used to make adjustments in order to achieve the optimum performance for each of the job classes.
Trestles – Supporting Science Gateways

- Gateways provide a scalable way to impact thousands of users. ~25% of TeraGrid users now access resources via gateways.

- Trestles supports Science Gateway users by using innovative scheduling approaches and also providing the middleware required for gateways. In particular:
  - Gateway users often have interactive expectations for gateway tools, so scheduling on Trestles is designed to support immediate runs for short jobs using "short-pool" reservations to hold nodes for near-future gateway jobs.
  - A single gateway may support a variety of usage models, e.g., both short- and long-running jobs. Both usage models are supported under Trestles.
  - The load imposed by gateways is unpredictable, and the Trestles scheduler supports flexible limits on the number of jobs that are allowed to execute simultaneously. This is important since the gateway use of a single community UNIX account sometimes reflects usage by thousands of users.
  - Gateways rely on the stability and availability of the underlying machines and on multiple levels of middleware like Globus, MyProxy, Condor and gridFTP to execute jobs remotely.
Data Oasis Heterogeneous Architecture
Lustre-based Parallel File System

- 3 Distinct Network Architectures
- Redundant Switches for Reliability and Performance
- 64 OSS (Object Storage Servers) Provide 100GB/s Performance and >4PB Raw Capacity
- JBODs (Just a Bunch Of Disks) Provide Capacity Scale-out to an Additional 5.8PB

Components:
- OSS 72TB
- JBOD 90TB
- MDS
- Arista 7508 10G
- TRESTLES IB cluster
- Mellanox 5020 Bridge 12 GB/s
- GORDON IB cluster
- 64 Lustre LNET Routers 100 GB/s
- TSCC cluster
- 10G Switch 25 GB/s
- Metal Data Servers
Data Oasis Performance

Lustre Performance to Gordon I/O Nodes

Bandwidth (MB/s)

Number of Lustre OSS Servers

Read
Write
## Which I/O system is right for my application?

<table>
<thead>
<tr>
<th></th>
<th>Flash-based I/O</th>
<th>Lustre</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Performance</strong></td>
<td>SSD’s support low latency I/O, high IOPS, and high bandwidth. One SSD can deliver 37K IOPS. Flash resources are dedicated to the user and performance is largely independent of what other users are doing on the system.</td>
<td>Lustre is ubiquitous in HPC. It does well for sequential I/O and files that support I/O to a few files from many cores simultaneously. Random I/O is a Lustre killer. Lustre is a shared resource and performance will vary depending on what other users are doing.</td>
</tr>
<tr>
<td><strong>Infrastructure</strong></td>
<td>On Gordon SSD’s are deployed in I/O nodes using iSER, an RDMA protocol that is accessed over the InfiniBand network. On Trestles the SSDs are locally available on compute nodes.</td>
<td>64 OSS’s; distinct file systems and metadata servers; accessed over a 10GbE network via the I/O nodes. Hundreds of HDDs/spindles.</td>
</tr>
<tr>
<td><strong>Persistence</strong></td>
<td>Data is generally removed at the end of a run so the resource can be made available to the next job.</td>
<td>Most is deployed as scratch and purgeable by policy (not necessarily at the end of the job. Some deployed as a persistent project storage resource.</td>
</tr>
<tr>
<td><strong>Capacity</strong></td>
<td>On Gordon: Up to 4.8 TB per users depending on configuration. On Trestles: 120 GB with 90GB available as local scratch.</td>
<td>No specific limits or quotas imposed on scratch. Largest file system is ~ 1.7 PB.</td>
</tr>
<tr>
<td><strong>Use cases</strong></td>
<td>Local application scratch (Abaqus, Gaussian); as a data mining platform (e.g., Hadoop); graph problems;</td>
<td>Traditional HPC I/O associated with MPI applications. Prestaging of data that will be pulled into flash.</td>
</tr>
</tbody>
</table>
HPC Resources at SDSC
Software Stack
**HPC Software Stack**

- Software managed primarily via rocks rolls.
- Wide range of libraries:

  **MPI Options (all built with gnu, intel, and pgi compilers):**
  - mvapich2
  - Openmpi
  - mpich2, mpich

  **Math / Scientific Libraries:**
  - MKL, ACML, Scalapack, Blas
  - GSL
  - SPRNG
  - SuperLU
  - FFTW
  - Metis/ParMetis

  **Data Libraries:**
  - HDF4, HDF5
  - NETCDF

  **Profiling/Debugging Tools:**
  - Valgrind, DDT, TAU, IPM
In addition to the libraries, a wide range of applications have been compiled and made available on the systems:

- MrBayes
- ABYSS
- Velvet
- AMBER
- NAMD
- GAMESS
- NWChem
- RAXML
- Garli
- Siesta
- Bowtie2
- fastq
Commercial Software

- Gaussian - Computational chemistry software. Site license covers NSF and UC resources.

- Abaqus – Multiphysics, modeling and meshing software.

- DDT – Debugging tool.

- VASP – Users with personal license can use it on our clusters. Compile support provided.

- Compilers – Intel, PGI.
HPC Applications at SDSC: Advanced Support

• Wide range of applications
  – Genomics
  – DNS of turbulent flow
  – Complex turbulent flow simulations for real world scenarios
  – Molecular dynamics simulations for drug design
  – Earthquake engineering simulations
  – Oceanographic simulations
  – Astrophysics simulations

• SDSC supports a wide range of high performance computing (HPC) scientific applications with its compute, I/O, and storage infrastructure.

• SDSC Staff are part of the Extended Collaborative Support Services (ECSS) program under the NSF XSEDE project. This program contributes to the computational science and engineering research by pairing academic researchers possessing exceptional scientific expertise with the computational expertise of ECSS staff in short to medium term collaborations.
Data Intensive Computing & Viz Stack

• Gordon was designed to enable data intensive computing.

• All clusters have access to the high speed lustre filesystem (Data Oasis: details in separate presentation) with an aggregated peak measured data rate of 100GB/s.

• Several libraries and packages have been installed to enable data intensive computing and visualization:
  • R – Software environment for statistical computing and graphics.
  • Weka – Tools for data analysis and predictive modeling
  • RapidMiner – Environment for machine learning, data mining, text mining, and predictive analytics
  • Octave
  • Matlab
  • VisIt
  • Paraview

• The myHadoop infrastructure was developed to enable use Hadoop for distributed data intensive analysis.
Hadoop: Application Areas

- Hadoop is widely used in data intensive analysis. Some application areas include:
  - Log aggregation and processing
  - Video and Image analysis
  - Data mining, Machine learning
  - Indexing
  - Recommendation systems

- Data intensive scientific applications can make use of the Hadoop MapReduce framework. Application areas include:
  - Bioinformatics and computational biology
  - Astronomical image processing
  - Natural Language Processing
  - Geospatial data processing

- Extensive list online at:
  - http://wiki.apache.org/hadoop/PoweredBy
Summary

• Diverse HPC and Data resources available – Gordon, Trestles, Data Oasis. TSCC cluster available for UCSD/UC users.

• SDSC provides multi-tiered support for scientific applications on HPC clusters. Starts with install of basic math, scientific, and data processing libraries and tools.

• Frequently used applications are pre-installed for users with optimizations for the HPC cluster architecture. Several commercial software packages available.

• Longer term collaborative support available for application development (via XSEDE ECSS).

• Science Gateways framework to allow researchers access via portals and avoid complex application set up and execution issues.