NERSC and HTC

Shane Canon, David Skinner and Jay Srinivasan
NUG2013

February 12, 2013
Science Strategies @ NERSC

Science at Scale
*Petascale to Exascale*

Science through Volume
*Thousands to Millions of Simulations*

Science in Data
*Petabytes to Exabytes*
MATERIALS PROJECT

A Materials Genome Approach

Accelerating materials discovery through advanced scientific computing and innovative design tools.

Database Statistics

19120 materials
214 intercalation batteries
4158 conversion batteries
3050 bandstructures

Press Highlights

The New York Times

Latest News
Common Themes

- Throughput Oriented / Embarrassingly parallel
- Rapidly Increasing demand for computation (outpacing Moore’s Law)
- Often Data Intensive
- Scaling from desktop or mid-range systems to HPC class systems
Approaches

• **Throughput Queues**

• **Private/User Allocation**
  – Task Farmer (NERSC Developed or Cray Provided)
  – MyHadoop
  – MySGE

• **Shared**
  – CCM/Torque

• **Hybrid?**
  – High-Throughput Queue Systems
• **Serial Queue on Carver**
  - 150 running
  - 20 eligible
  - Best for serial jobs needing full Linux stack

• **Throughput Queue on Hopper**
  - 250 running
  - 500 eligible
  - Best for high-throughput, small concurrency jobs
TaskFarmer

Server
- Portable
- Reads in query genes
- Tracks progress and re-runs failed tasks
- Maintains checkpoint
- Collects output from clients

Client
- Can run any executable or script
- Gathers command line arguments from server
- Fetches input from server and pushes back results
Strengths of MapReduce and Hadoop

• Fault Tolerance Model
• Data Locality
• Simple Programming Model
• Hides Complexity
• Domain Specific Extensions
• Strong Community
• User submits a single parallel job
• Personnel SGE scheduler is started
• User can submit jobs to SGE without modifications
• User still needs to think about scaling issues
Downside to Private Approach

- Load imbalance can lead to wasted resources and additional charging
- Other users can’t take advantage of idle cores
Running a shared-node Serial workload on the XE-6 using CCM
Using CCM to run a shared-node serial workload

- CCM can be used to “convert” XE-6 (MPP) compute nodes into standard “cluster-like” nodes with a regular Linux environment.
- To run a serial workload on these “CCM nodes” requires they be accessible as regular cluster nodes to the batch system
  - This cannot be done using the regular batch system
  - This requires starting up a separate batch system instance
  - Done using a special CCM “job” which starts up the server and client daemons – the server is started up on the standard XE-6 MOM nodes, and the clients are on the XE-6 CCM compute nodes
Mechanics of running a shared-node serial workload

- “Special” user submits a job to the ccm_queue, asking for as many nodes as required to handle a serial workload (subject to CCM limits), and for the maximum time allowed.
- “Special job” starts up pbs_server on XE-6 MOM node with alternate ports
- Job then runs pbs_mom on allocated CCM compute nodes (under alternate ports)
- Job starts up scheduler (Maui or pbs_sched) which communicates with the alternate resource manager (RM)
- At this point, other users (user1, user2, etc) can submit jobs to the CCM compute nodes (which have now been essentially repurposed as a separate cluster supporting a serial workload)
<table>
<thead>
<tr>
<th>Job ID</th>
<th>Username</th>
<th>Queue</th>
<th>Jobname</th>
<th>SessionID</th>
<th>NDS</th>
<th>TSK</th>
<th>Memory</th>
<th>Req'd Time</th>
<th>Elap Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>nid00002</td>
<td>jay</td>
<td>serial</td>
<td>tst.job</td>
<td>22129</td>
<td>--</td>
<td>1</td>
<td>--</td>
<td>00:10</td>
<td>00:04</td>
</tr>
<tr>
<td>nid00002</td>
<td>jay</td>
<td>serial</td>
<td>tst.job</td>
<td>22132</td>
<td>--</td>
<td>1</td>
<td>--</td>
<td>00:10</td>
<td>00:04</td>
</tr>
<tr>
<td>nid00002</td>
<td>jay</td>
<td>serial</td>
<td>tst.job</td>
<td>22135</td>
<td>--</td>
<td>1</td>
<td>--</td>
<td>00:10</td>
<td>00:04</td>
</tr>
<tr>
<td>nid00002</td>
<td>jay</td>
<td>serial</td>
<td>tst.job</td>
<td>22138</td>
<td>--</td>
<td>1</td>
<td>--</td>
<td>00:10</td>
<td>00:04</td>
</tr>
<tr>
<td>nid00002</td>
<td>jay</td>
<td>serial</td>
<td>tst.job</td>
<td>22153</td>
<td>--</td>
<td>1</td>
<td>--</td>
<td>00:10</td>
<td>00:04</td>
</tr>
<tr>
<td>nid00002</td>
<td>jay</td>
<td>serial</td>
<td>tst.job</td>
<td>22199</td>
<td>--</td>
<td>1</td>
<td>--</td>
<td>00:10</td>
<td>00:03</td>
</tr>
<tr>
<td>nid00002</td>
<td>jay</td>
<td>serial</td>
<td>tst.job</td>
<td>22233</td>
<td>--</td>
<td>1</td>
<td>--</td>
<td>00:10</td>
<td>00:03</td>
</tr>
<tr>
<td>nid00002</td>
<td>jay</td>
<td>serial</td>
<td>tst.job</td>
<td>22284</td>
<td>--</td>
<td>1</td>
<td>--</td>
<td>00:10</td>
<td>00:03</td>
</tr>
<tr>
<td>nid00002</td>
<td>jay</td>
<td>serial</td>
<td>tst.job</td>
<td>26151</td>
<td>--</td>
<td>1</td>
<td>--</td>
<td>00:10</td>
<td></td>
</tr>
<tr>
<td>nid00002</td>
<td>jay</td>
<td>serial</td>
<td>tst.job</td>
<td>26166</td>
<td>--</td>
<td>1</td>
<td>--</td>
<td>00:10</td>
<td></td>
</tr>
<tr>
<td>nid00002</td>
<td>jay</td>
<td>serial</td>
<td>tst.job</td>
<td>26186</td>
<td>--</td>
<td>1</td>
<td>--</td>
<td>00:10</td>
<td></td>
</tr>
<tr>
<td>nid00002</td>
<td>jay</td>
<td>serial</td>
<td>tst.job</td>
<td>26198</td>
<td>--</td>
<td>1</td>
<td>--</td>
<td>00:10</td>
<td></td>
</tr>
<tr>
<td>nid00002</td>
<td>jay</td>
<td>serial</td>
<td>tst.job</td>
<td>26239</td>
<td>--</td>
<td>1</td>
<td>--</td>
<td>00:10</td>
<td></td>
</tr>
<tr>
<td>nid00002</td>
<td>jay</td>
<td>serial</td>
<td>tst.job</td>
<td>26288</td>
<td>--</td>
<td>1</td>
<td>--</td>
<td>00:10</td>
<td></td>
</tr>
<tr>
<td>nid00002</td>
<td>jay</td>
<td>serial</td>
<td>tst.job</td>
<td>26332</td>
<td>--</td>
<td>1</td>
<td>--</td>
<td>00:10</td>
<td></td>
</tr>
<tr>
<td>nid00002</td>
<td>jay</td>
<td>serial</td>
<td>tst.job</td>
<td>26394</td>
<td>--</td>
<td>1</td>
<td>--</td>
<td>00:10</td>
<td></td>
</tr>
<tr>
<td>JOBNAME</td>
<td>USERNAME</td>
<td>STATE</td>
<td>PROC</td>
<td>REMAINING</td>
<td>STARTTIME</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>----------</td>
<td>---------</td>
<td>------</td>
<td>-----------</td>
<td>-------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>canon</td>
<td>Running</td>
<td>1</td>
<td>00:00:00</td>
<td>Mon Apr 30 09:36:34</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>canon</td>
<td>Running</td>
<td>1</td>
<td>00:00:00</td>
<td>Mon Apr 30 09:36:34</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>canon</td>
<td>Running</td>
<td>1</td>
<td>00:00:00</td>
<td>Mon Apr 30 09:36:34</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>canon</td>
<td>Running</td>
<td>1</td>
<td>00:00:00</td>
<td>Mon Apr 30 09:36:34</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>canon</td>
<td>Running</td>
<td>1</td>
<td>00:00:00</td>
<td>Mon Apr 30 09:36:34</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>canon</td>
<td>Running</td>
<td>1</td>
<td>00:00:00</td>
<td>Mon Apr 30 09:36:34</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>canon</td>
<td>Running</td>
<td>1</td>
<td>00:00:00</td>
<td>Mon Apr 30 09:36:34</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>canon</td>
<td>Running</td>
<td>1</td>
<td>00:00:00</td>
<td>Mon Apr 30 09:36:34</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>jay</td>
<td>Running</td>
<td>1</td>
<td>00:08:58</td>
<td>Mon Apr 30 09:35:32</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>jay</td>
<td>Running</td>
<td>1</td>
<td>00:08:58</td>
<td>Mon Apr 30 09:35:32</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>jay</td>
<td>Running</td>
<td>1</td>
<td>00:08:58</td>
<td>Mon Apr 30 09:35:32</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>jay</td>
<td>Running</td>
<td>1</td>
<td>00:08:58</td>
<td>Mon Apr 30 09:35:32</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>jay</td>
<td>Running</td>
<td>1</td>
<td>00:08:58</td>
<td>Mon Apr 30 09:35:32</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>jay</td>
<td>Running</td>
<td>1</td>
<td>00:08:58</td>
<td>Mon Apr 30 09:35:32</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>jay</td>
<td>Running</td>
<td>1</td>
<td>00:08:58</td>
<td>Mon Apr 30 09:35:32</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>jay</td>
<td>Running</td>
<td>1</td>
<td>00:08:58</td>
<td>Mon Apr 30 09:35:32</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

16 Active Jobs 16 of 24 Processors Active (66.67%)
1 of 1 Nodes Active (100.00%)

<table>
<thead>
<tr>
<th>JOBNAME</th>
<th>USERNAME</th>
<th>STATE</th>
<th>PROC</th>
<th>WCLIMIT</th>
<th>QUEUETIME</th>
</tr>
</thead>
</table>

0 Idle Jobs

Total Jobs: 16   Active Jobs: 16   Idle Jobs: 0   Blocked Jobs: 0
Limitations

- **Current approach uses a static assignment of nodes.**
  - Initial request for CCM nodes needs cannot be changed on the fly, but multiple requests can be made

- **CCM communication occurs over TCP/IP, so the high-performance network is not available. (Can’t share uGNI)**
Other Challenges

• Policy and Fairness – Many of the challenges result from policies, not just technical
• I/O and Staging Common Files – Python or Perl libraries, large common references, etc
• Porting – Moving applications to Cray can be difficult
• Still not like a Cluster – No local disk, limited networking
Future Work

• **Continue to Improve CCM/Torque Approach**
  – Finish testing and phase into production
  – Dynamically resize serial partition

• **Improve Hadoop Implementation**
  – Optimize shuffle phase for high-bandwidth network

• **Evaluating more fine-grained tasked based Scheduler**
  – Use of external message queue (i.e. AMQP)
Closing Thoughts

• Increasing demand to support new workloads
  – Driven by improving instruments
  – New classes of modeling and simulation

• NERSC has developed four approaches to supporting new workloads and is exploring others
Why does NERSC Support This?

• Users need it
• Important science can be achieved
• Accelerate specific analysis
• Small fraction of a large system is significantly larger than available systems
• Even “Capability” jobs often have through-put oriented components (pre-computing, analysis)
Map/Array Job

Large Input

Filter/Map → Output
Filter/Map → Output
Filter/Map → Output
Filter/Map → Output
Map/Reduce

Large Input

Filter/Map
Filter/Map
Filter/Map
Filter/Map

Shuffle

Reduce
Reduce
Reduce
Reduce

Output
Complex Workflows

Large Input

Filter/Map  Filter/Map  Filter/Map  Filter/Map

Reorder

2nd Step  2nd Step  2nd Step  2nd Step

Finalize
26