INTRODUCTION TO BELL

Xiao Zhu– Senior Research Scientist, Research Computing
Bell is named in honor of **Prof. Clara Bell Sessions**, a trailblazer on minority rights and Professor and Director of Continuing Education of Nursing

- Established the Minority Student Nurses' Association
- Led the formation of Purdue’ Minority Faculty Fellows program
- Co-chair of National Congress of Black Faculty Council on Research and Education (1992, 1993)
- Cabinet member on the Human Rights Committee of the American Nurses Association
- Charter member of the Association of Black Nursing Faculty in Higher Education
2020 Cluster - Bell

- **448 Nodes - 2.1 PF** (2x 2013’s Conte)
- **AMD EPYC 7662 processors**: 128 core nodes (2.0 GHz base, 3.3 GHz boost), 256 GB RAM per node
- 100 Gpbs HDR Infiniband
- Direct-to-chip liquid cooling
- 5 PB Lustre Scratch (100 TB flash)
  - 50% faster than Brown’s storage
- 8 Large memory nodes (1 TB RAM)
- GPU subsystem based on AMD MI50 GPUs (16 GPUs)
- Composable Kubernetes infrastructure for non-batch workflows
- **6 year cluster life**

*Early User Access started on 10/16/2020*
AMD EPYC 7662 Processor Architecture

- 128 cores per node!
  - 64 cores/socket, dual sockets
  - 8 Core Complex Dies (CCDs)
  - 2 CCX per CCD, 8 cores per CCD
  - CCDs connect to RAM, I/O, and each other through I/O dies

- Supporting AVX2
AMD EPYC 7662 Processor Architecture

- Faster memory: DDR 4 memory at 3200 MHz

- Cache Advantage:
  - 16MB share L3 per CCX
  - 128 MB L3 cache
  - 4.5MB of L2 + L3 cache per core

- PCIe Gen4
  - Up to 128 lanes of high speed I/O
  - 100Gbps HDR Infiniband per node
Setting thread/process placement is important!
NUMA – Non-Uniform Memory Access
Four NUMA domains per socket is the typical HPC configuration and what is used on Bell
lstopo-no-graphics or numactl -H
Costs and Details

- Price point – 64-core shares for $4,000
  - Compares very favorably to Halstead’s 20 cores for $3,600

- Base nodes are equivalent to the bulk of Snyder
  - Snyder will not be expanded further
  - Going forward, the very-large memory need will be met by the shared queue with 1+ TB RAM nodes

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Dollars per GF</th>
<th>Dollars per Core</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steele</td>
<td>$34.42</td>
<td>$251.75</td>
</tr>
<tr>
<td>Coates</td>
<td>$21.83</td>
<td>$218.25</td>
</tr>
<tr>
<td>Rossmann</td>
<td>$15.25</td>
<td>$139.21</td>
</tr>
<tr>
<td>Hansen</td>
<td>$13.27</td>
<td>$122.13</td>
</tr>
<tr>
<td>Carter</td>
<td>$10.10</td>
<td>$206.25</td>
</tr>
<tr>
<td>Conte</td>
<td>$2.90</td>
<td>$418.75</td>
</tr>
<tr>
<td>Rice</td>
<td>$4.62</td>
<td>$220.00</td>
</tr>
<tr>
<td>Halstead</td>
<td>$3.78</td>
<td>$180.00</td>
</tr>
<tr>
<td>Brown</td>
<td>$2.86</td>
<td>$233.29</td>
</tr>
<tr>
<td>Bell</td>
<td>$1.57</td>
<td>$57.81</td>
</tr>
</tbody>
</table>
Bell Cluster

User Experience Changes

- The 128-core Rome CPUS are the top-end, for low-end pricing! (We usually get mid-tier parts)
- Resource-specific home directory
- Moving away from password-based auth
  - BoilerKey two-factor
  - SSH Keys
- Specialized nodes available to all for the price of entry
  - Large memory nodes in a shared queue
  - GPU nodes in a shared queue
Access

- **SSH**
  - `bell.rcac.purdue.edu`

- **Remote Desktop**
  - `desktop.bell.rcac.purdue.edu`
  - Web browser, Thinlinc client

- **Gateway (available soon)**
  - `gateway.bell.rcac.purdue.edu`
  - Access to compute node directly from browser

- **Jupyter Notebook / Rstudio**
  - Supported on Gateway
  - Jupyter Notebook or Rstudio sessions on compute nodes
Software

- Compilers
  - gcc 9.3.0-default compiler, 10.2.0, (-march=znver2)
  - gcc 4.8.5, 6.3.0
  - Intel compilers 17.0.1, 19.0.5 (-march=core-avx2, -xHost)
  - AOCC 2.1 (-march=znver2)

- MPI libraries
  - OpenMPI (default 3.1.4)
  - IntelMPI
  - Mvapich2 (coming soon)

- Recommended stack
Software

- **Math libraries**
  - Intel MKL, OpenBlas, AOCL
  - If your application heavily uses Intel MKL routines, setting the following environment variable is beneficial:
    
    ```
    export MKL_DEBUG_CPU_TYPE=5
    ```
  - When using FFTW interface from MKL, please also set:
    
    ```
    export MKL_CBWR=AUTO
    ```

- **Python**
  - Anaconda 3.7, 3.8
  - Natively built Numpy, Scipy (coming)

- **R**
  - 3.6.3, 4.0.0
  - Built with gcc/9.3.0, MKL
Bell supports a wide array of applications and libraries.

<table>
<thead>
<tr>
<th>Application Packages &amp; Libraries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math</td>
</tr>
<tr>
<td>I/O libraries</td>
</tr>
<tr>
<td>CFD</td>
</tr>
<tr>
<td>Material Modeling</td>
</tr>
<tr>
<td>Chemistry</td>
</tr>
<tr>
<td>Machine Learning</td>
</tr>
<tr>
<td>Structural Mechanics</td>
</tr>
<tr>
<td>Visualization</td>
</tr>
<tr>
<td>Debugging and Profiling</td>
</tr>
<tr>
<td>Bioinformatics</td>
</tr>
</tbody>
</table>
Files

- **Home**
  - Dedicated to Bell
  - If you have data on other clusters, use globus or “copy-rcac-home”
  - [https://www.rcac.purdue.edu/knowledge/bell/storage/transfer/copyhome](https://www.rcac.purdue.edu/knowledge/bell/storage/transfer/copyhome)

- **Scratch**
  - Lustre parallel filesystem
  - 200 TB, 2M files per user
  - 60 day purge policy

- **Fortress**
  - Use Globus
  - Use HSI: [https://www.rcac.purdue.edu/knowledge/bell/faq/data/transferkeytab](https://www.rcac.purdue.edu/knowledge/bell/faq/data/transferkeytab)
    - Need to copy keytab from other clusters (`scp -pr ~/.private bell:`)
    - Only Bell access: log into “data.rcac.purdue.edu”, run “hsi”, and copy keytab to Bell
STREAM

<table>
<thead>
<tr>
<th>Application</th>
<th>Performance on Skylake</th>
<th>Performance on Rome</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB12</td>
<td>510.0</td>
<td>1600.0</td>
</tr>
<tr>
<td>HPL</td>
<td>1518.9</td>
<td>3403.5</td>
</tr>
<tr>
<td>HPCG</td>
<td>25.5</td>
<td>40.2</td>
</tr>
<tr>
<td>STREAM(^b)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COPY</td>
<td>144064.5</td>
<td>203522.5</td>
</tr>
<tr>
<td>SCALE</td>
<td>143006.6</td>
<td>196341.9</td>
</tr>
<tr>
<td>ADD</td>
<td>148988.5</td>
<td>215621.8</td>
</tr>
<tr>
<td>TRIAD</td>
<td>148831.5</td>
<td>223098.8</td>
</tr>
</tbody>
</table>

a. Unit for each DB12 entry is Monte Carlo events/s, units for each HPCG/HPL entry is GFLOPs and unit for each STREAM entry is MB/s.

b. Results when using all the cores are chosen. The all-core memory bandwidth is the more accurate representation of the memory bandwidth for majority of HPC/HTC workloads.
BELL - BENCHMARKS

![Bar chart showing improvement in applications/benchmarks for Brown and Rome](chart.png)

- Applications/Benchmarks:
  - GROMACS
  - LAMMPS
  - NAMD
  - R
  - VASP
  - OPENFOAM
  - WRF
  - DB12
  - HPL
  - HPCG

- Improvement scale:
  - 0
  - 1
  - 2
  - 3
  - 4
  - 5

- Comparison between Brown and Rome.
BELL - BENCHMARKS

- HPL
- HPCG
- GROMACS
- NAMD
- OPENFOAM
- WRF
- VASP

Scaling vs. # of cores.
THANK YOU

rcac.purdue.edu/purchase