Introduction to
Using the Ada Cluster

T. Mark Huang

HPRC Short Course – 2015 Summer
Be a good compute citizen ...

- It is illegal to share computer passwords and accounts by state law and university regulation;
- It is prohibited to use Ada in any manner that violates the United States export control laws and regulations, EAR & ITAR;
- Abide by the expressed or implied restrictions in using commercial software
- Also, only home directory is backup daily.
Hardware Summary

- https://sc.tamu.edu/wiki/index.php/Ada:Intro
Ada – an x86 Cluster

A 17,500-core system, with 845 20-core nodes equipped with the INTEL 10-core 2.5GHz IvyBridge processor. The other 15 nodes are 1TB and 2TB memory, 4-processor SMPs configured with the INTEL 10-core 2.7GHz Westmere processor. 30 nodes have 2 GPUs each and 9 have 2 Phi coprocessors. The interconnecting fabric is based on the Mellanox SX6536 IB core switch (the middle rack in the graphic above).
Ada Schematic: 17,460-core 858-node Cluster

IB Fabric: Based on the Mellanox SX6536 Core Switch

792 NeXtScale

20 – Core IvyBridge Node
64 GB DRAM

20 – Core IvyBridge Node
64 GB DRAM

15 EXTRA-LARGE MEMORY NODES

40 – Core Westmere Node
1TB DRAM

40 – Core Westmere Node
2TB DRAM

11 1TB + 4 2TB

8 Login

20 – Core IvyBridge Node @2 GPUs/PHIs
256 GB DRAM

43 iDataPlex Nodes

20 – Core IvyBridge Node @2 GPUs
256/64 GB DRAM

20 – Core IvyBridge Node @2 PHIs
64 GB DRAM

30 @2GPU + 9 @2PHI + 4 @ no accel (256GB)

Disc

IBM GSS26
4PB
Sep 2014

ETHERNET LAN
Access and File Transfer

- SSH only
  - On campus: ssh NetID@ada.tamu.edu
  - Off campus:
    - Set up VPN: http://hdc.tamu.edu/Connecting/VPN/
    - Then: ssh NetID@ada.tamu.edu
  - PuTTY (Windows) or MobaXTerm (Windows; also with X11)
  - Login nodes enforce idle process timeout (60 minutes)

- File Transfer:
  - scp: command line (Linux, MacOS)
  - WinSCP: GUI (Windows)
  - FileZilla: GUI (Windows, MacOS, Linux)

- Bulk data transfer: use FTN nodes with GridFTP, Globus Connect, bbcp

- https://sc.tamu.edu/wiki/index.php/Ada:Filesystems_and_Files#Transferring_Files
File Systems and Quota

- https://sc.tamu.edu/wiki/index.php/Ada:Filesystems_and_Files
Computing Environments

• Paths:
  – PATH (/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/netid/bin)
  – LD_LIBRARY_PATH

• Many applications, many versions, and many paths ....... How do you manage them?!

• Modules
  – Help: module help
  – Search: module spider [package_name]
  – Load: module load package1 package2 ... (case sensitive)
  – List: module list
  – Unload: module unload package1 package2 ...
  – Purge: module purge

• https://sc.tamu.edu/wiki/index.php/Ada:Computing_Environment#Modules
Life cycle of a job

1. **Internet**
2. **VPN + SSH**
3. **Login nodes**
4. **Job file**
5. **Queue**
6. **LSF (batch manager)**
7. **Output**
8. **Cluster**

**SSH**

**Login nodes** to **Cluster** via **LSF** (batch manager) and **Queue**.
Job Files (1)

#BSUB -J myjob1  # sets the job name to myjob1.
#BSUB -L /bin/bash  # uses the bash login shell to initialize the job's execution environment
#BSUB -W 12:30  # sets to 12.5 hours the job's runtime wall-clock limit.
#BSUB -n 1  # assigns 1 core for execution.
#BSUB -o stdout1.%J  # directs the job's standard output to stdout1.jobid

##

# <--- at this point the current working directory is the one you submitted the job from.
#
# module load intel  # loads the INTEL software tool chain to provide, among other things,
#                   # needed runtime libraries for the execution of prog.exe below.
#                   # (assumes prog.exe was compiled using INTEL compilers.)
#
# prog.exe < input1 >& data_out1  # both input1 and data_out1 reside in the job submission dir
##

- https://sc.tamu.edu/wiki/index.php/Ada:Batch#Job_files
Job Files (2)

#BSUB -J myjob2
# sets the job name to myjob1.
#BSUB -L /bin/bash
# uses the bash login shell to initialize the job's execution environment.
#BSUB -W 12:30
# sets to 12.5 hours the job's runtime wall-clock limit.
#BSUB -n 3
# assigns 3 cores for execution.
#BSUB -R "span[ptile=3]"
# assigns 3 cores per node.
#BSUB -R "rusage[mem=5000]"
# reserves 5000MB per process/CPU for the job (i.e., 15,000 MB for job/node)
#BSUB -M 5000
# sets to 5,000MB (~5GB) the per process enforceable memory limit.
#BSUB -o stdout2.%J
# directs the job's standard output to stdout2.jobid
#BSUB -P project1
# charges the consumed service units (SUs) to project1.
#BSUB -u e-mail_address
# sends email to the specified address
#BSUB -B -N
# send emails on job begin (-B) and end (-N)

##

cd $SCRATCH/myjob2
# makes $SCRATCH/myjob2 the job's current working directory
module load intel
# loads the INTEL software tool chain to provide, among other things,

# The next 3 lines concurrently execute 3 instances of the same program, prog.exe, with standard input and output data streams assigned to different files in each case.

(prog.exe < input1 >& data_out1 ) &
(prog.exe < input2 >& data_out2 ) &
(prog.exe < input3 >& data_out3 )

wait

• https://sc.tamu.edu/wiki/index.php/Ada:Batch#Job_files
Submit Job and Job Tracking

• Submit a job: `bsub < jobfile1`

• Job tracking:
  - `bjobs [-u all or user_name] [-l] job_id`
  - `bpeek [-f] job_id`
  - `bkill job_id`
  - `bhist [-l] job_id`

• [https://sc.tamu.edu/wiki/index.php/Ada:Batch#Job_tracking_and_control_commands](https://sc.tamu.edu/wiki/index.php/Ada:Batch#Job_tracking_and_control_commands)
Environment Variables

- $SCRATCH = /scratch/user/NetID
- https://sc.tamu.edu/wiki/index.php/Ada:Batch#Environment_Variables
Important Job Parameters

#BSUB -n NNN

# NNN: total number of cores/jobslots to allocate for the job

#BSUB -R "span[ptile=XX]"

# XX: number of cores/jobslots per node to use

#BSUB -R "select[node-type]"

# node-type: nxt, mem256gb, gpu, phi, mem1t, mem2t ...

#BSUB -R "rusage[mem=nnn]"

# reserves nnn MBs per process/CPU for the job

#BSUB -M nnn

# sets the per process enforceable memory limit to nnn MB

#BSUB -W hh:mm or mm

# sets job's runtime wall-clock limit in hours:minutes or just minutes
Job slots - ptile

#BSUB –n 10 –R “span[ptile=2]” ...

will allocate 10 jobslots, 2 per node. That is, the job will span 5 nodes;

#BSUB –n 80 –R “span[ptile=20]” ...

will allocate 4 whole nodes (80/20), not including the X-large memory ones.
Request memory

- Requesting memory via
  
  ```bash
  #BSUB -R "rusage[mem=process_alloc_size]"
  #BSUB -M process_size_limit
  ```

- Default 2.5 GB per process (core) if -R/-M not specified, but it might cause memory contention when sharing a node with other jobs.

- On NeXTScale (nxt) nodes, usable memory is at best 55 GB (out of total 64 GB). That means per process limit should not exceed 2800 MB for a 20-core job.

- If more memory is needed, use large memory nodes, mem256gb (20 cores), mem1tb (40 cores), or mem2tb (40 cores). mem1tb and mem2tb nodes are accessible via `xlarge` queue.
### Queues (1)

#### $ bqueues $

<table>
<thead>
<tr>
<th>QUEUE_NAME</th>
<th>PRIO</th>
<th>STATUS</th>
<th>MAX</th>
<th>JL/U</th>
<th>JL/P</th>
<th>JL/H</th>
<th>NJOBS</th>
<th>PEND</th>
<th>RUN</th>
<th>SUSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>staff</td>
<td>150</td>
<td>Open:Active</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>special</td>
<td>120</td>
<td>Open:Active</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>4380</td>
<td>0</td>
<td>4380</td>
<td>0</td>
</tr>
<tr>
<td>xlarge</td>
<td>100</td>
<td>Open:Active</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>40</td>
<td>0</td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td>vnc</td>
<td>90</td>
<td>Open:Active</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>devel</td>
<td>80</td>
<td>Open:Active</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>short</td>
<td>70</td>
<td>Open:Active</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>120</td>
<td>0</td>
<td>120</td>
<td>0</td>
</tr>
<tr>
<td>medium</td>
<td>60</td>
<td>Open:Active</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>18520</td>
<td>11740</td>
<td>6780</td>
<td>0</td>
</tr>
<tr>
<td>long</td>
<td>55</td>
<td>Open:Active</td>
<td>6000</td>
<td>2000</td>
<td>-</td>
<td>-</td>
<td>6074</td>
<td>1636</td>
<td>4438</td>
<td>0</td>
</tr>
<tr>
<td>general</td>
<td>50</td>
<td>Open:Active</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>small</td>
<td>50</td>
<td>Open:Active</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>751</td>
<td>232</td>
<td>519</td>
<td>0</td>
</tr>
<tr>
<td>curie_devel</td>
<td>40</td>
<td>Open:Active</td>
<td>32</td>
<td>32</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>curie_medium</td>
<td>35</td>
<td>Open:Active</td>
<td>512</td>
<td>192</td>
<td>-</td>
<td>-</td>
<td>384</td>
<td>256</td>
<td>128</td>
<td>0</td>
</tr>
<tr>
<td>curie_long</td>
<td>30</td>
<td>Open:Active</td>
<td>192</td>
<td>32</td>
<td>-</td>
<td>-</td>
<td>32</td>
<td>0</td>
<td>32</td>
<td>0</td>
</tr>
<tr>
<td>curie_general</td>
<td>25</td>
<td>Open:Active</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>training</td>
<td>1</td>
<td>Open:Active</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
### Queues (2)

<table>
<thead>
<tr>
<th>Queue</th>
<th>Min/Default/Max Cpus</th>
<th>Default/Max Walltime</th>
<th>Compute Node Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>special</td>
<td>None</td>
<td>1 hr / 36 hr</td>
<td>All</td>
</tr>
<tr>
<td>devel</td>
<td>1 / 1 / 320</td>
<td>10 min / 1 hr</td>
<td>All</td>
</tr>
<tr>
<td>small</td>
<td>1 / 1 / 3</td>
<td>1 hr / 120 hr</td>
<td>64 GB and 256 GB nodes</td>
</tr>
<tr>
<td>short</td>
<td>4 / 4 / 8000</td>
<td>1 hr / 5 hr</td>
<td>64 GB and 256 GB nodes</td>
</tr>
<tr>
<td>medium</td>
<td>4 / 4 / 4000</td>
<td>5 hr / 24 hr</td>
<td>64 GB and 256 GB nodes</td>
</tr>
<tr>
<td>long</td>
<td>4 / 4 / 2000</td>
<td>24 hr / 7 days</td>
<td>64 GB and 256 GB nodes</td>
</tr>
<tr>
<td>xlarge</td>
<td>1 / 1 / 280</td>
<td>1 hr / 200 hr</td>
<td>1 TB nodes (11), 2 TB nodes (4)</td>
</tr>
<tr>
<td>vnc</td>
<td>1 / 1 / 20</td>
<td>1 hr / 6 hr</td>
<td>All 30 nodes with GPUs</td>
</tr>
</tbody>
</table>

- [https://sc.tamu.edu/wiki/index.php/Ada:Batch#Queues](https://sc.tamu.edu/wiki/index.php/Ada:Batch#Queues)
OpenMP Job File

• Must set `OMP_NUM_THREADS` to take advantage of all cores
• All processes run on the same node. Request mem1tb or mem2tb if you need up to 40 cores per node

```bash
#BSUB -n 20 -R 'rusage[mem=300] span[ptile=20]' -M 300
#BSUB -J omp_helloWorld
#BSUB -o omp_helloWorld.%J -L /bin/bash -W 20
#
module load intel
#
ifort -openmp -o omp_helloWorld.exe omp_helloWorld.f90
#
export OMP_NUM_THREADS=20
./omp_helloWorld.exe
```
MPI Job File

- Processes may be run on multiple nodes

```
#BSUB -n 12 -R 'rusage[mem=150] span[ptile=4]' -M 150
#BSUB -J mpi_helloWorld -o mpi_helloWorld.%J -L /bin/bash -W 20
#
module load intel
#
mpiifort -o mpi_helloWorld.exe mpi_helloWorld.f90
#
mpiexec.hydra -np 12 ./mpi_helloWorld.exe
```
Other Type of Jobs

- GPU
- PHI
- vnc
Service Units

- New AMS (Account Management System) will be online for testing on July 1st.
- Each SU is corresponding to 1 hour of wall clock time elapsed on 1 effective CPU core. Effective CPU core depends on number of cores, amount of memory, and type of nodes requested. Final formula will be announced.
- SU balance message after a job is submitted (testing; ignore for now)
Compiler and Running Programs

- https://sc.tamu.edu/wiki/index.php/Ada
Common mistakes / problems

- DOS/Windows edited job files (^M): `dos2unix jobfile`
- Did not load required module(s)
- Missing program, input files
- Insufficient walltime
- Insufficient memory
- No matched resource (-R rusage[mem] too large)
- OpenMP jobs across nodes
- Disk quota
- Use GUI without setup X11 forwarding
- License availability
tamulauncher

- https://sc.tamu.edu/wiki/index.php/Ada:Tamulauncher
Remote Visualization

• X11 forwarding

• VNC jobs
  - https://sc.tamu.edu/wiki/index.php/Ada:Remote_Visualization
How to ask for Help

- Check Documents/Wiki (https://sc.tamu.edu/wiki/index.php/Ada) for possible solutions first.
- Email your questions to help@sc.tamu.edu.
- Help us, help you -- we need more info
  - Which Cluster
  - UserID/NetID
  - Job id(s) if any
  - Location of your jobfile, input/output files
  - Application used if any
  - Module(s) loaded if any
  - Error messages
  - Steps you have taken, so we can reproduce the problem
- Or visit us @ 104 Teague. Making an appointment is recommended.