Argonne Training Program on Extreme-Scale Computing (ATPESC)

Quick Start on ATPESC Computing Resources

Ray Loy
ATPESC 2017 Deputy Program Director

Q Center, St. Charles, IL (USA)
Date 07/30/2017
OUTLINE

• ALCF Systems
  – KNL (Theta)
  – Blue Gene/Q (Mira, Cetus, Vesta)
  – x86+GPU (Cooley)

• OLCF
  – Cray XK7 (Titan)

• NERSC
  – KNL (Cori)
The DOE Leadership Computing Facility

- Collaborative, multi-lab, DOE/SC initiative ranked top national priority in *Facilities for the Future of Science: A Twenty-Year Outlook*.

- Mission: Provide the computational and data science resources required to solve the most important scientific & engineering problems in the world.

- Highly competitive user allocation program (INCITE, ALCC).

- Projects receive 100x more hours than at other generally available centers.

- LCF centers partner with users to enable science & engineering breakthroughs (Liaisons, Catalysts).
### Leadership Computing Facility System

<table>
<thead>
<tr>
<th></th>
<th>Argonne LCF</th>
<th>Oak Ridge LCF</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System</strong></td>
<td>IBM Blue Gene/Q</td>
<td>Cray XK7</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Mira</td>
<td>Titan</td>
</tr>
<tr>
<td><strong>Compute nodes</strong></td>
<td>49,152</td>
<td>18,688</td>
</tr>
<tr>
<td><strong>Node architecture</strong></td>
<td>PowerPC, 16 cores</td>
<td>AMD Opteron, 16 cores</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NVIDIA K20x (Kepler) GPU</td>
</tr>
<tr>
<td><strong>Processing Units</strong></td>
<td>786,432 Cores</td>
<td>299,008 x86 Cores +</td>
</tr>
<tr>
<td></td>
<td></td>
<td>18,688 GPUs</td>
</tr>
<tr>
<td><strong>Memory per node, (gigabytes)</strong></td>
<td>16</td>
<td>32 + 6</td>
</tr>
<tr>
<td><strong>Peak performance, (petaflops)</strong></td>
<td>10</td>
<td>27</td>
</tr>
</tbody>
</table>
ALCF Systems

- **Mira – BG/Q**
  - 49,152 nodes / 786,432 cores
  - 786 TB of memory
  - Peak flop rate: 10 PetaFLOPs
  - 3,145,728 hardware threads

- **Vesta (T&D) - BG/Q**
  - 2,048 nodes / 32,768 cores

- **Cetus (debug) - BG/Q**
  - 4,096 nodes / 65,536 cores

- **Cooley (visualization & data analysis) – Cray CS**
  - 126 nodes, each with
    - Two Intel Xeon E5-2620 Haswell 2.4 GHz 6-core processors
    - NVIDIA Tesla K80 graphics processing unit with 24 GB memory
    - 384 GB DDR4 memory

- **Storage**
  - Scratch: 28.8 PB raw capacity, 240 GB/s bw (GPFS)
  - Home: 1.8 PB raw capacity, 45 GB/s bw (GPFS)

- **Theta - Cray XC40**
  - 3,624 nodes / 231,935 cores
ALCF KNL (Theta)
Theta

Theta is the first of two pre-exascale systems coming to Argonne

- Serves as a bridge between Mira and Aurora, transition and data analytics system
- Cray XC40 system. Runs Cray software stack
- 9.65 PF peak performance
- 3624 nodes with 2nd Generation Intel® Xeon Phi™ processor
  - codename Knights Landing, 7230 SKU 64 cores 1.3GHz
- 192GB DDR4 memory 16GB MCDRAM on each node
- 128GB SSD on each node
- Cray Aries high speed interconnect in dragonfly topology
- Initial file system: 10PB Lustre file system, 200 GB/s throughput
- Currently in T2O phase running ESP & ADSP projects
- Iota - 44 node, air cooled TD system
Filesystems

- GPFS
  - Home directories (/home) are in /gpfs/mira-home
    - Default quota 50GiB
    - Your home directory is backed up

- Lustre
  - Project directories (/projects) are in /lus/theta-fs0/projects (e.g. /projects/ATPESC2017)
    - Access controlled by unix group of your project
    - Default quota 1TiB
    - NOT backed up
  - With large I/O, be sure to consider stripe width
Modules (Theta ONLY)

- A tool for managing a user’s environment
  - Sets your PATH to access desired front-end tools
  - Your compiler version can be changed here

- module commands
  - help
  - list ← what is currently loaded
  - avail
  - load
  - unload
  - switch\swap
  - use ← add a directory to MODULEPATH
  - display\show
Compilers

- For all compilers (Intel, Cray, Gnu, etc):
  - **Use**: cc, CC, ftn
  - **Do not use** mpicc, MPICC, mpic++, mpif77, mpif90
    - *they do not generate code for the compute nodes*
- Selecting the compiler you want using "module swap" or "module unload" followed by "module load"
  - Intel
    - PrgEnv-intel  *This is the default*
  - Cray
    - module swap PrgEnv-intel PrgEnv-cray
    - **NOTE**: links libsci by default
  - Gnu
    - module swap PrgEnv-intel PrgEnv-gnu
  - Clang/LLVM
    - module swap PrgEnv-intel PrgEnv-llvm
Theta Job script

#!/bin/bash
#COBALT -t 10
#COBALT -n 2
#COBALT -A Myprojectname

# Various env settings are provided by Cobalt
echo $COBALT_JOBDID $COBALT_PARTNAME $COBALT_JOBSIZE

aprun -n 16 -N 8 -d 1 -j 1 -cc depth ./a.out
status=$?

# could do another aprun here...

exit $status
Aprun overview

- **Options**
  - `-n total_number_of_ranks`
  - `-N ranks_per_node`
  - `-d depth` [number of cpus (hyperthreads) per rank]
  - `-cc depth` [Note: depth is a keyword]
  - `-j hyperthreads` [cpus (hyperthreads) per compute unit (core)]

- **Env settings you may need**
  - `-e OMP_NUM_THREADS=nthreads`
  - `-e KMP_AFFINITY=…`

- See also **man aprun**
Submitting a Cobalt job

- qsub -A <project> -q <queue> -t <time> -n <nodes> ./jobscript.sh
  E.g.
  qsub -A Myprojname -q cache-quad t -t 10 -n 32 ./jobscript.sh

- If you specify your options in the script via #COBALT, then just:
  - qsub jobscript.sh

- Make sure jobscript.sh is executable
- Without "-q", submits to the queue named "default"
- Without "-A", uses environment variable COBALT_PROJ if set
  - export COBALT_PROJ=ATPESC2017
- man qsub for more options

**Theta has 8 node minimum job size**
Managing your job

- qstat – show what's in the queue
  - `qstat -u <username>` # Jobs only for user
  - `qstat <jobid>` # Status of this particular job
  - `qstat -fl <jobid>` # Detailed info on job

- `qdel <jobid>`

- showres – show reservations currently set in the system

- `man qstat` for more options
Cobalt files for a job

- Cobalt will create 3 files per job, the basename `<prefix>` defaults to the jobid, but can be set with “qsub -O myprefix”
  - jobid can be inserted into your string e.g. "-O myprefix_${jobid}"

- **Cobalt log file**: `<prefix>.cobaltlog`
  - created by Cobalt when job is submitted, additional info written during the job
  - contains submission information from qsub command, runjob, and environment variables

- **Job stderr file**: `<prefix>.error`
  - created at the start of a job
  - contains job startup information and any content sent to standard error while the user program is running

- **Job stdout file**: `<prefix>.output`
  - contains any content sent to standard output by user program
Interactive job

- Useful for short tests or debugging
- Submit the job with –I (letter I for Interactive)
  - Default queue and default project
    - qsub –I –n 32 –t 30
  - Specify queue and project:
    - qsub –I –n 32 –t 30 –q training –A ATPESC2017
- Wait for job's shell prompt
  - This is a new shell with env settings e.g. COBALT_JOBID
  - Exit this shell to end your job
- From job's shell prompt, run just like in a script job, e.g.
  - aprun -n 512 -N 16 -d 1 -j 1 -cc depth ./a.out
- After job expires, apruns will fail. Check qstat $COBALT_JOBID
Core files and debugging

- Abnormal Termination Processing (ATP)
  - Set environment `ATP_ENABLED=1` in your job script before `aprun`
  - On program failure, generates a merged stack backtrace tree in file `atpMergedBT.dot`
  - View the output file with the program `stat-view` (module load `stat`)

- Notes on linking your program
  - PrgEnv-cray links everything necessary by default
  - PrgEnv-intel
    - May need to link with `-Wl,-T/opt/cray/pe/cce/8.5.2/craylibs/x86-64/2.23.1.cce.ld`

- Other debugging tools
  - You can generate STAT snapshots asynchronously
  - Full-featured debugging with DDT
  - More info at
    - https://collab.cels.anl.gov/display/ESP/ATP+and+STAT
    - https://collab.cels.anl.gov/display/ESP/Allinea+Forge+(DDT)
ALCF Blue Gene/Q (Mira, Cetus, Vesta)
Similar to **modules** package

- Keys are read at login time to set environment variables like PATH.
  - Mira, Cetus, Vesta: ~/.soft
  - Cooley: ~/.soft.cooley

To get started:

```
# This key selects XL compilers to be used by mpi wrappers
+mpiwrapper-xl
@default

# the end - do not put any keys after the @default
```

- After edits to .soft, type "resoft" or log out and back in again
Using compiler wrappers

- **IBM XL cross-compilers:**
  - SoftEnv key: +mpiwrapper-xl
  - Non-thread-safe: mpixlc, mpixlcxx, mpixlf77, mpixlf90, mpixlf95, mpixlf2003, etc.
  - **Thread-safe** (add _r suffix): mpixlc_r, mpixlcxx_r, mpixlf77_r, etc.
  - Example: mpixlc –O3 –o hellompi hellompi.c

- **GNU cross-compilers:**
  - SoftEnv key: +mpiwrapper-gcc
  - mpicc, mpicxx, mpif77, mpif90

- **CLANG cross-compilers:**
  - SoftEnv key: +mpiwrapper-bgclang
  - mpiclang, mpiclang++, mpiclang++11

http://www.alcf.anl.gov/user-guides/software-and-libraries
BG/Q Job script

- Sample:

```bash
#!/bin/bash
#COBALT -n 32 -t 30 -q training -A ATPESC2017
# -p is mode (how many ranks per node)
# --np is number of ranks
runjob -p 16 --np 32 --block $COBALT_PARTNAME : hellompi
# Note: exit status of this script is runjob's status
```

- Some args use **single** dash and some **double** dash (**man runjob**)
- Don't forget --block. COBALT_PARTNAME is set automatically by Cobalt.
- You can do multiple runjobs in succession
  - Use normal shell redirection to separate output
- Must use --envs to pass environment variables into your program
- Output to <jobid>.{output,error,cobaltlog} (use -O to change prefix)
Submitting your job

- `qsub -A <project> -q <queue> -t <time> -n <nodes> --mode script ./jobscript.sh`
  
  E.g.
  
  `qsub -A ATPESC2017 -q training -t 10 -n 32 --mode script ./jobscript.sh`

  *Note: runs on Mira should use "default" queue*

- If you specify your options in the script via `#COBALT`, then just:
  - `qsub jobscript.sh`
  - Make sure `jobscript.sh` is executable
  - Without "-q", submits to the queue named "default"
  - Without "-A", uses environment variable `COBALT_PROJ` if set
    - `export COBALT_PROJ=ATPESC2017`
  - `man qsub` for more options
Interactive job

- Useful for short tests or debugging
- Submit the job with –l (letter I for Interactive)
  - Default queue and default project
    - qsub –l –n 32 –t 30
  - For the workshop:
    - qsub –l –n 32 –t 30 –q training –A ATPESC2017
- Wait for job's shell prompt
  - *This is a new shell* with settings COBALT_PARTNAME, COBALT_JOBID
  - Exit this shell to end your job
- **Run "wait-boot" \(\leftarrow\)** Important!
- From job's shell prompt, run just like in a script job:
- After job expires, runjob will fail. Check qstat $COBALT_JOBID
About node count and mode

- **Node count**
  - Minimum physical partition sizes available depend on machine
    - Vesta: 32  Cetus: 128  Mira: 512
    - Your job will get the smallest available size >= what you ask for
      - It is reserved for you; you are charged for entire partition

- **Mode**
  - How many MPI ranks per node
    - Possible values: 1,2,4,8,16,32,64
  - A node has 16 cores, each can run 4 threads
    - For modes < 16, an MPI rank will be assigned more than one core
    - Example: "-p 4" can run up to 16 threads per MPI rank
Using OpenMP

- Shared-memory parallelism is supported within a single node
  - Use MPI across compute nodes, OpenMP within a compute node
- **For XL compilers, thread-safe compiler version should be used** (mpixlc_r etc.) with any threaded application (either OMP or Pthreads)
  - OpenMP standard directives are supported (version 3.1)
  - Compile with –qsmp=omp,noauto  (Note: debugging use noopt)
  - Increase default thread stack size using environment value XLSMPOPTS=stack=NNN  (value per thread, e.g. 10M)
- Setting number of OpenMP threads
  - set using environment variable OMP_NUM_THREADS
  - must be exported to the compute nodes using runjob –envs

**Example:** 32 nodes / 512 ranks / 4 threads per rank:

```bash
#!/bin/bash
#COBALT –n 32 –t 30
runjob –block $COBALT_PARTNAME –p 16 --np 512 --envs OMP_NUM_THREADS=4 : a.out
```
ALCF x86+GPU (Cooley)
Cooley Job Script

- More like a typical Linux cluster
- Job script different than BG/Q.
  - Example test.sh:
    ```sh
    #!/bin/sh
    NODES=`cat $COBALT_NODEFILE | wc -l`
    PROCS=$(($NODES * 12))
    mpirun -f $COBALT_NODEFILE -n $PROCS myprog.exe
    ```
  - Submit on 5 nodes for 10 minutes
    ```bash
    qsub -n 5 -t 10 -A ATPESC2017 ./test.sh
    ```
  - Refer to online user guide for more info
ATPESC Resources

ALCF – Mira, Cetus, Vesta, Cooley, and Theta

- **Project name:** ATPESC2017
- **Note:** use your ALCF Username. The password will be your old/newly established PIN + token code displayed on the token.
- **Support:** on-site ALCF staff available to help you!! and support@alcf.anl.gov
- **Reservations:** Please check the details of the reservations directly on each machine (**command**: showres)
- **Queue:** training
Cryptocard tips

- The displayed value is a hex string. Type your PIN followed by all letters as CAPITALS.
- If you fail to authenticate the first time, you may have typed it incorrectly
  - Try again with the same crypto string (do NOT press button again)
- If you fail again, try a different ALCF host with a fresh crypto #
  - A successful login resets your count of failed logins
- Too many failed logins → your account locked
  - Symptom: You get password prompt but login denied even if it is correct
- Too many failed logins from a given IP → the IP will be blocked
  - Symptom: connection attempt by ssh or web browser will just time out
Machine status web page

http://status.alcf.anl.gov/mira/activity (a.k.a. The Gronkulator)
References

• Sample files
  – On Theta
    • /projects/ATPESC2017/examples/getting-started
  – On Vesta, Mira, Cetus, or Cooley:
    • /projects/ATPESC2016/examples/getting-started

• Online docs
  – www.alcf.anl.gov/user-guides
Additional resources: OLCF and NERSC
ATPESC Resources

OLCF – Titan

- **Project name:** GEN007_atpesc

- **Note:** use the Username printed on the envelope the token came in. It will be csep01, csep02, etc. The password will be your newly established PIN + token code displayed on the token.

- **Support:** [help@olcf.anl.gov](mailto:help@olcf.anl.gov) or call 1-865-241-6536

- See documents in your Argonne Folder for additional information
ATPESC Resources

NERSC – Edison and Cori

- **Project name:** ntrain
- **Note:** `ssh machine_name.nersc.gov`
- **Support:** [accounts@nersc.gov](mailto:accounts@nersc.gov) or call 1-800-666-3772

Questions?

• *Use this presentation as a reference during ATPESC!*

• Supplemental info will be posted as well
## DOE ASCR Computing Upgrades At a Glance

<table>
<thead>
<tr>
<th>System attributes</th>
<th>NERSC Now</th>
<th>OLCF Now</th>
<th>ALCF Now</th>
<th>NERSC Upgrade</th>
<th>OLCF Upgrade</th>
<th>ALCF Upgrades</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name Planned Installation</td>
<td>Edison</td>
<td>Titan</td>
<td>Mira</td>
<td>Cori 2016</td>
<td>Summit 2017-2018</td>
<td>Theta 2016</td>
</tr>
<tr>
<td>System peak (PF)</td>
<td>2.6</td>
<td>27</td>
<td>10</td>
<td>&gt; 30</td>
<td>150</td>
<td>&gt;8.5</td>
</tr>
<tr>
<td>Peak Power (MW)</td>
<td>2</td>
<td>9</td>
<td>4.8</td>
<td>&lt; 3.7</td>
<td>10</td>
<td>1.7</td>
</tr>
<tr>
<td>Total system memory</td>
<td>357 TB</td>
<td>710TB</td>
<td>768TB</td>
<td>~1 PB DDR4 + High Bandwidth Memory (HBM)+1.5PB persistent memory</td>
<td>&gt;1.74 PB DDR4 + HBM + 2.8 PB persistent memory</td>
<td>&gt;480 TB DDR4 + High Bandwidth Memory (HBM)</td>
</tr>
<tr>
<td>Node performance (TF)</td>
<td>0.460</td>
<td>1.452</td>
<td>0.204</td>
<td>&gt; 3</td>
<td>&gt; 40</td>
<td>&gt; 3</td>
</tr>
<tr>
<td>Node processors</td>
<td>Intel Ivy Bridge</td>
<td>AMD Opteron Nvidia Kepler</td>
<td>64-bit PowerPC A2</td>
<td>Intel Knights Landing many core CPUs Intel Haswell CPU in data partition</td>
<td>Multiple IBM Power9 CPUs &amp; multiple Nvidia Voltas GPUS</td>
<td>Intel Knights Landing Xeon Phi many core CPUs</td>
</tr>
<tr>
<td>System size (nodes)</td>
<td>5,600 nodes</td>
<td>18,688 nodes</td>
<td>49,152</td>
<td>9,300 nodes 1,900 nodes in data partition</td>
<td>~3,500 nodes</td>
<td>&gt;2,500 nodes</td>
</tr>
<tr>
<td>System Interconnect</td>
<td>Aries</td>
<td>Gemini</td>
<td>5D Torus</td>
<td>Aries</td>
<td>Dual Rail EDR-IB</td>
<td>Aries</td>
</tr>
<tr>
<td>File System</td>
<td>7.6 PB 168 GB/s, Lustre®</td>
<td>32 PB 1 TB/s, Lustre®</td>
<td>26 PB 300 GB/s GPFS™</td>
<td>28 PB 744 GB/s Lustre®</td>
<td>120 PB 1 TB/s Lustre®</td>
<td>10PB, 210 GB/s Lustre® initial</td>
</tr>
</tbody>
</table>
Memory Modes - IPM and DDR
Selected at node boot time

- Two memory types
  - In Package Memory (IPM)
    - 16 GB MCDRAM
    - ~480 GB/s bandwidth
  - Off Package Memory (DDR)
    - Up to 384 GB
    - ~90 GB/s bandwidth
- One address space
- Possibly multiple NUMA domains
- Memory configurations
  - Cached: DDR fully cached by IPM
    - Flat: user managed
  - Hybrid: ¼, ½ IPM used as cache
- Managing memory:
  - jemalloc & memkind libraries
  - Pragmas for static memory allocations
Theta queues and modes

- MCDRAM and NUMA modes can only be set by the system when nodes are rebooted. *Users cannot directly reboot nodes.*

- Submit job with the --attrs flag to get the mode you need. E.g.
  - `qsub -n 32 -t 60 --attrs mcdram=cache:numa=quad ./jobscript.sh`

- Other mode choices
  - mcdram: cache, flat, split, equal
  - numa: quad, a2a, hemi, snc2, snc4

- Queues
  - Normal jobs use queue named "default"
  - Debugging: debug-cache-quad, debug-flat-quad
    - Note: pre-set for mcdram/numa configuration
  - "qstat -Q" lists all queues