Argonne NATIONAL LABORATORY

Crash Course: Running MPI Programs on the ALCF Blue Gene/Q

ATPESC
August 3, 2014

Ray Loy
Applications Performance Engineering
ALCF

U.S. DEPARTMENT OF ENERGY

References

- Sample files
  - {vesta,cetus,mira}:~rloy/public/atpesc2014
- Online docs
  - www.alcf.anl.gov/user-guides

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 be blocked
  - Symptom: connection attempt by ssh or web browser will just time out

Softenv

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

Job script

- Sample:

```
#!/bin/bash
#COBALT –n 32 –t 30 –q Q.ATPESC –A ATPESC2014
# -p is mode (how many ranks per node)
# --np is number of ranks
runjob –p 16 --np 32 --block $COBALT_PARTNAME : hellompi
return 0
```

- 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
- Use --envs to add environment variables
- 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 ATPESC2014 –q Q.ATPESC –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=ATPESC2014
- **man qsub** for more options

Managing 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

Interactive job

- Useful for short tests or debugging
- Submit the job with –I
  - Default queue and default project
    - qsub –I –n 32 –t 30
  - For the workshop:
    - qsub –I –n 32 –t 30 –q Q.ATPESC –A ATPESC2014
- Wait for job's shell prompt
  - *This is a new shell* with settings COBALT_PARTNAME, COBALT_JOBID
  - Exit this shell to end your job
- From job's shell prompt, run just like a script job:
  - runjob –block $COBALT_PARTNAME –p 16 –np 32 : hellompi
- After job expires, runjob will fail. ***Check*** **qstat $COBALT_JOBID**

Access to computing resources

- ALCF resources
  - Vesta -- 2-rack Blue Gene/Q (one rack dedicated to ATPESC 24/7 tonight through August 9; both racks available during scheduled time slots if desirable)
  - Cetus -- 4-rack Blue Gene/Q
  - Tukey -- visualization cluster with NVDIA GPUs)
  - Mira (as time allows)
- Arrangements for using Edison at NERSC will be done today
- Arrangements for using Titan at OLCF will be later in the program

ALCF resources for ATPESC

- Vesta will be the main resource for ATPESC jobs
  - run your jobs on Vesta unless larger nodecounts/longer walltimes are necessary
  - queue limits are similar to those of the default queue: 1hr walltime and 1024 node-hours max, maximum of 2 running jobs and 10 queued jobs
- Default queues will be stopped an hour before scheduled hands-on sessions in the afternoon and evening, and started again afterwards.
- You will be able to submit and run jobs on Vesta outside of scheduled hands-on periods, but will be competing with users in the default queue for resources
- Cetus will be used for students with
  - a) greater ability to scale, and
  - b) who wish to run larger/longer jobs during scheduled hands-on sessions.
- Avoid using Cetus for jobs less than 128 nodes in size
- Cetus has a max partition size of 2048. following the evening hands-on session
- No Q.ATPESC queue on Mira

ALCF resources for ATPESC

- Vesta - just submit to Q.ATPESC
- Cetus - just submit to Q.ATPESC;
  - you may have to wait in queue a bit during the afternoon
- Mira - submit to the default queue and notify a BG/Q admin

About node count and mode

- Node count
  - Minumum 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
  - 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:

```
#!/bin/bash
#COBALT –n 32 –t 30
runjob –block $COBALT_PARTNAME –p 16 --np 512 --envs OMP_NUM_THREADS=4 : a.out
```

Hands-on