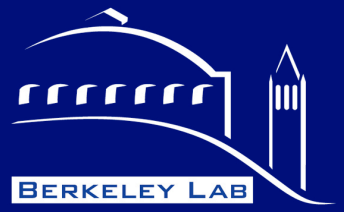
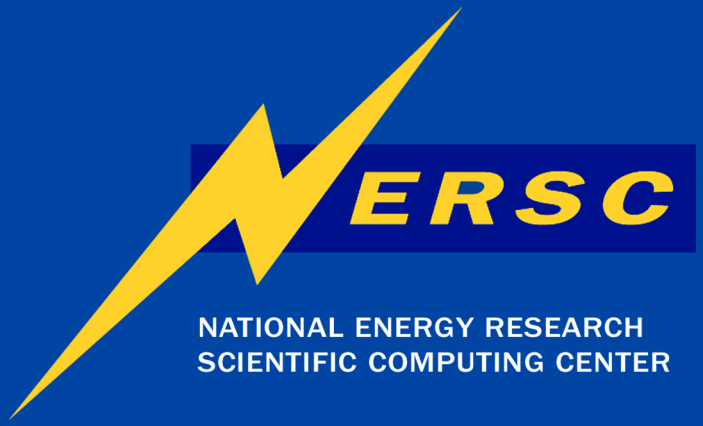




Submitting and Running Jobs on the Cray XT5

Richard Gerber
NERSC User Services
RAGerber@lbl.gov

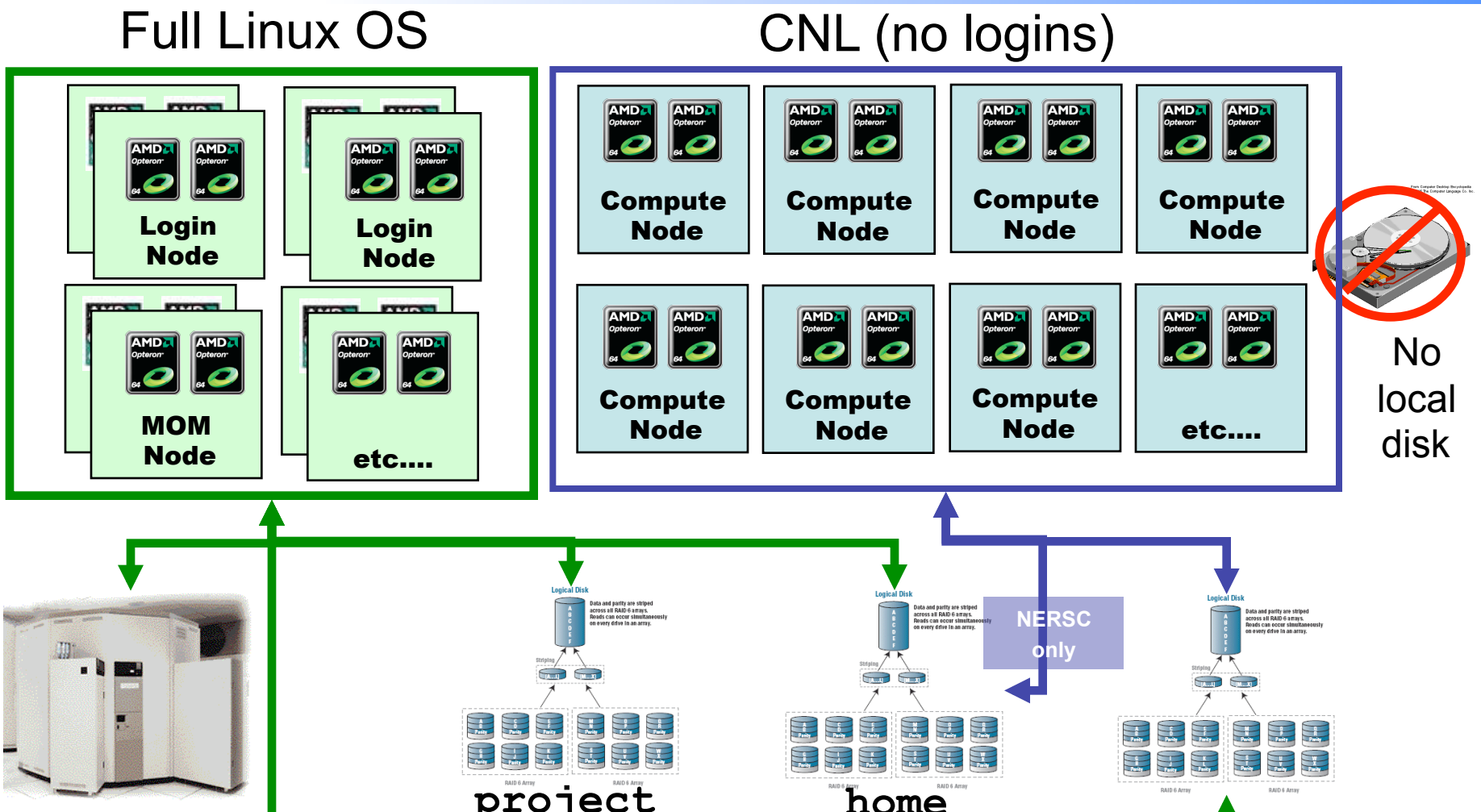
Joint Cray XT5 Workshop
UC-Berkeley
February 1, 2010

Outline

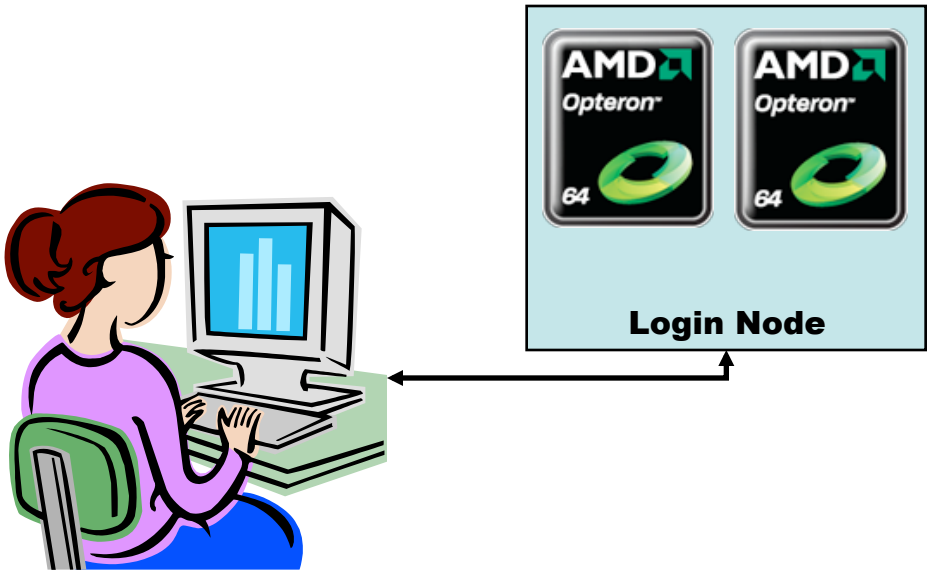
Hopper in blue; Jaguar in Orange; Kraken in Green

- XT5 Overview
- Creating and Submitting a Batch Job
- How a Job Is Launched
- Monitoring Your Job
- Queues and Policies

Cray XT5 Overview



Running a Job on the XT5



Login nodes run a full version of Linux

1. Log in from your desktop using SSH
2. Compile your code or load a software module
3. Write a job script
4. Submit your script to the batch system
5. Monitor your job's progress
6. Archive your output
7. Analyze your results

Outline

- Cray XT5 Overview
- Creating and Submitting a Batch Job
- How a Job Is Launched
- Monitoring Your Job
- NERSC Queues and Policies

Job Scripts

A job script is a text file. Create and edit with a text editor, like vi or emacs.

Directives specify how to run your job

UNIX commands run on a service node (Full Linux)

code.x runs in parallel on compute nodes

```
#PBS -l walltime=01:00:00
#PBS -l mppwidth=4096
#PBS -l mppnppn=8
#PBS -l size=4096
#PBS -q regular
#PBS -N BigJob
#PBS -V
#PBS -A mp999

cd $PBS_O_WORKDIR

echo "Starting at" `date`

aprun -n 4096 -N 8 ./code.x
```

Common Directives

Hopper

```
#PBS -q queue_name
```

Specify the *queue* in which to run.

```
#PBS -l walltime=HH:MM:SS
```

Specify the max *wallclock time* your job will use.

```
#PBS -M email_address
```

Specify the email address for notifications.

```
#PBS -V
```

Copy your current environment into batch environment.

```
#PBS -A account
```

Charge job to *account*.

Specifying Job Size

The fundamental schedulable unit on the XT5 is a *compute node*.

Your Torque/PBS directives tell the system how many compute nodes to reserve for your job.

You have exclusive access to a compute node. You “own” every node that is allocated for your job for the duration of your job.

Specifying Job Size

Hopper

```
#PBS -l mppwidth=number_of_instances
```

Set mppwidth equal to the total number of copies of your executable to run in parallel.

```
#PBS -l mppnppn=instances_per_node
```

Set mppnppn equal to the # of instances to run per node.

Jaguar

Kraken

```
#PBS -l size=cores
```

Set size equal to the # of cores that will be available for your job to use. Must be a multiple of 12

Sample Hopper Batch Script

Hopper

```
#!/bin/bash -l
#PBS -q debug
#PBS -l mppwidth=384
#PBS -l mppnppn=8
##### NOTE: 48 nodes requested
#PBS -l walltime=00:30:00
#PBS -N myFirstTest
#PBS -M my_email@my_school.edu
#PBS -V

cd $PBS_O_WORKDIR


aprun -n 384 -N 8 ./a.out
```

Sample Jaguar/Kraken Script

Jaguar

Kraken

```
#/bin/bash -l
#PBS -l size=384
##### NOTE: 384/12=32 nodes requested
#PBS -l walltime=00:30:00
#PBS -N myFirstTest
#PBS -M my_email@my_school.edu
#PBS -V

cd $PBS_O_WORKDIR

aprun -n 384 ./a.out
```

Running N tasks per node

Note that you never directly specify the number of nodes.

It is implicit in your settings for mppwidth and mppnppn or size.

You may want to run fewer tasks (instances) per node than there are cores per node to increase the memory available per MPI task.

Hopper

```
#PBS -l mppwidth=512
#PBS -l mppnppn=2
```

This will allocate 256 nodes to run 512 tasks using 2 tasks per node. (Must be consistent with aprun options; see below)

Jaguar Kraken

```
#PBS -l size=768
```

This will always allocate 64 nodes (768/12). You will use aprun to control tasks/node (see below).

Submitting Jobs

Submit your job script with the **qsub** command.

```
nid04100% qsub script_name
```

The batch script directives (#**PBS –whatever**) can be specified on the qsub command line. For example:

```
nid04100% qsub -A account script_name
```

Use `-A account (or repo)` to specify the account to charge.

I recommend putting everything you care about explicitly in the batch script to avoid ambiguity and to have a record of exactly how you submitted your job.

Modifying Jobs

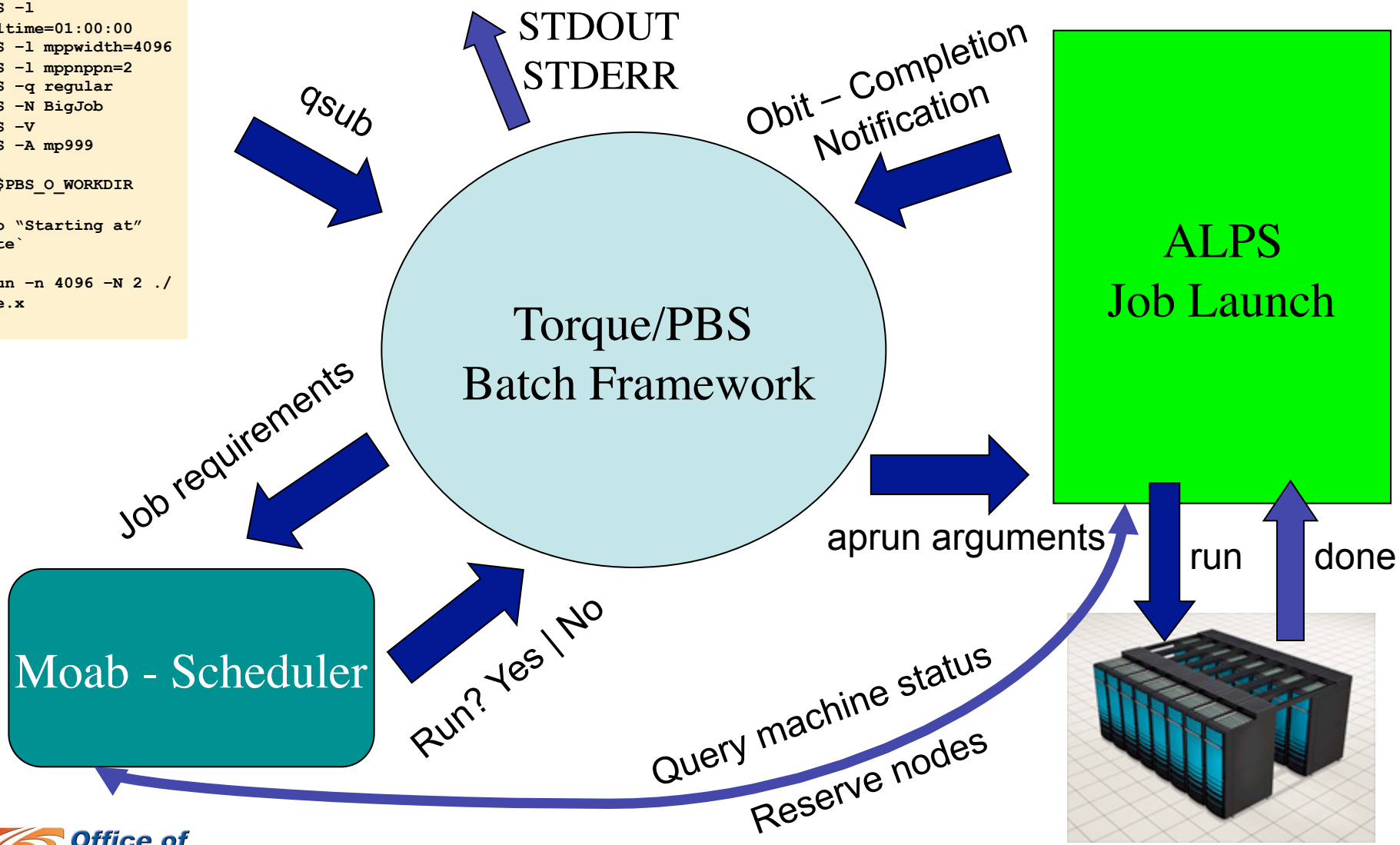
- qdel <jobid>: deletes queued job
- qhold <jobid>: holds job in queue
- qrls <jobid>: release held job
- qalter <jobid> <options>
  - You can modify some parameters
  - See "`man qalter`"

Outline

- XT5 Overview
- Creating and Submitting a Batch Job
- How a Job Is Launched
- Monitoring Your Job
- Queues and Policies

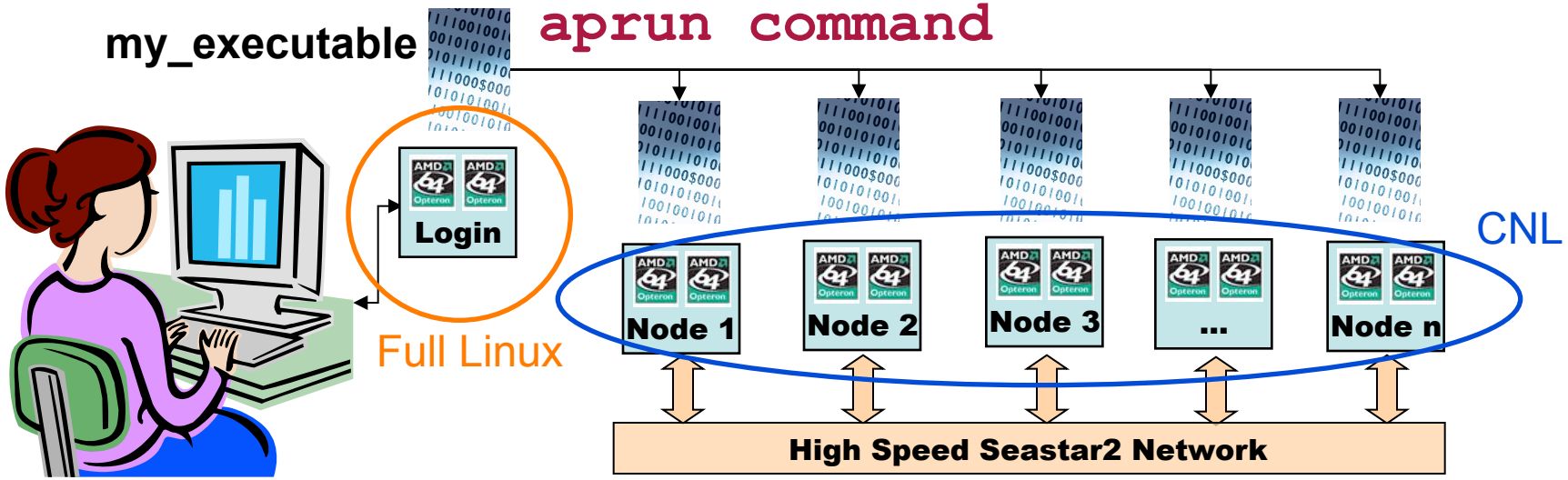
Job Scheduling and Launch

```
#PBS -l
walltime=01:00:00
#PBS -l mppwidth=4096
#PBS -l mppnppn=2
#PBS -q regular
#PBS -N BigJob
#PBS -V
#PBS -A mp999

cd $PBS_O_WORKDIR

echo "Starting at"
`date`

aprun -n 4096 -N 2 ./
code.x
```



Parallel Job Launch - ALPS

ALPS = Application Level Placement Scheduler




```
aprun -n instances my_executable
```

For MPI-only codes, this is equivalent to

```
aprun -n mpi_tasks my_executable
```

aprun options for XT5

You use options to aprun to tell ALPS how to run your job

Tasks per compute node

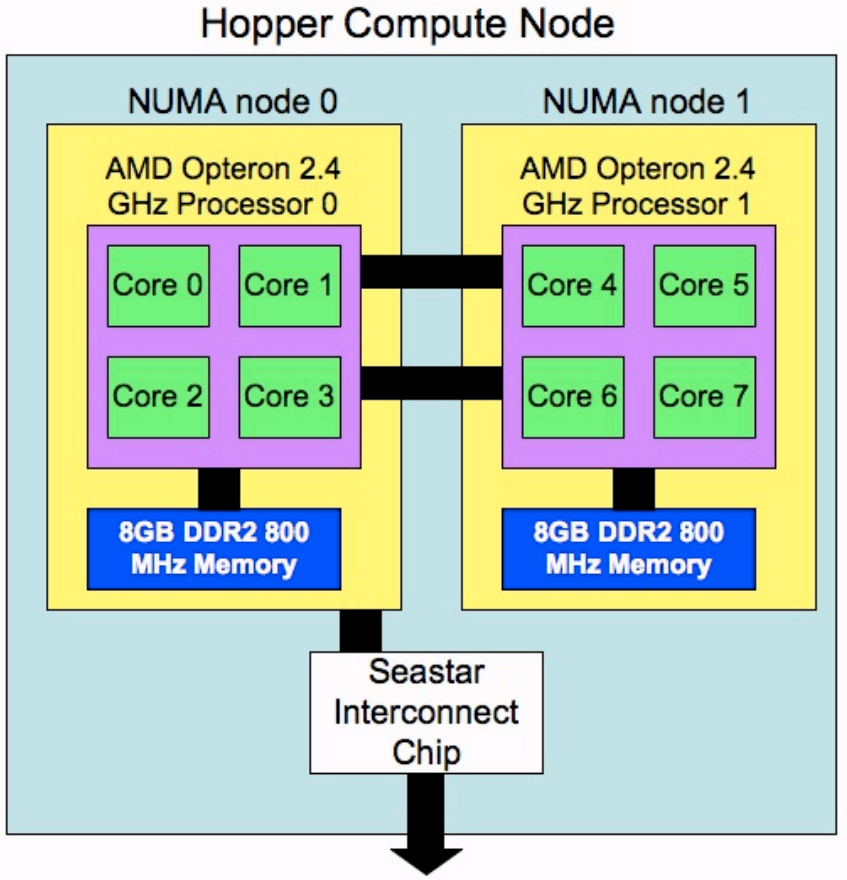
```
-N [ 1-8 ] Hopper
-N [ 1-12 ] Jaguar, Kraken
```

Cores per Opteron

```
-S [ 1-4 ] Hopper
-S [ 1-6 ] Jaguar, Kraken
```

Opterons (or NUMA nodes, sockets) per compute node

```
-sn [ 1-2 ]
```



Terminology: CPU = core
1 Opteron fits in 1 socket

aprun affinity options

CPU affinity

`-cc[ cpu | numa_node | none ]`

cpu: task bound to 1 core (default)

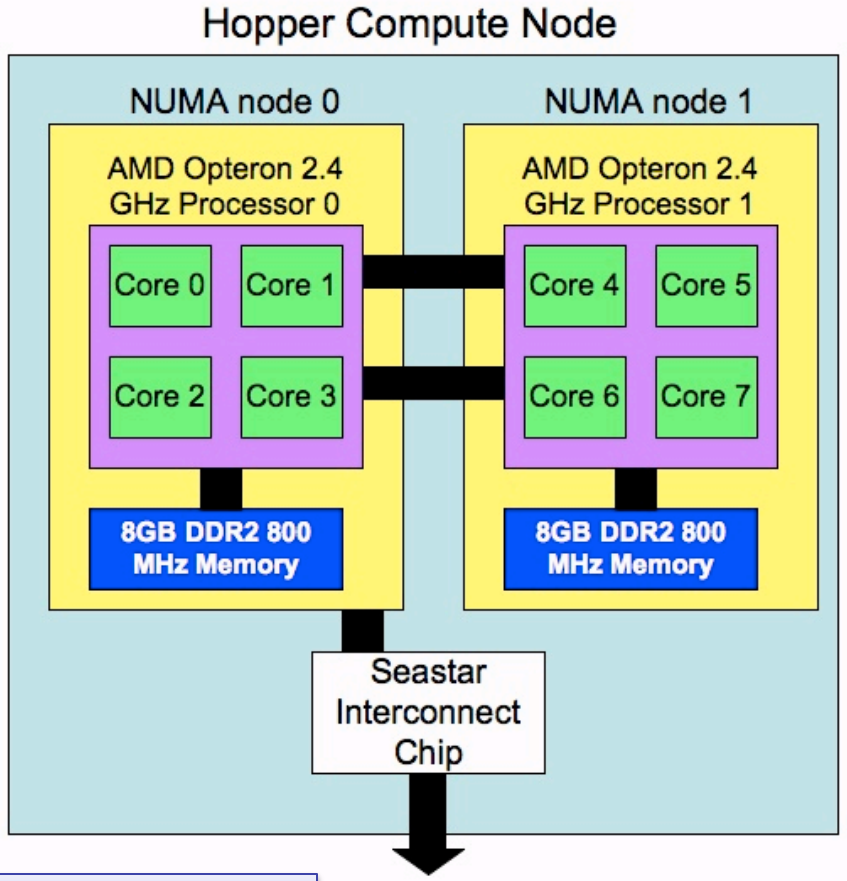
numa_node: task bound 1 Opteron

none: task can migrate to both

Memory affinity

`-ss`

Restrict memory access to within a NUMA node. Default: not set.



Defaults are fine in most cases; see “man aprun”

PBS Directives vs. aprun options

Hopper

aprun –n # must be consistent with
#PBS –l mppwidth=#

Ditto for –N and
#PBS –l mppnppn

Mismatches will cause job launch errors

```
#PBS -l walltime=01:00:00
#PBS -l mppwidth=4096
#PBS -l mppnppn=8
#PBS -q regular
#PBS -N BigJob
#PBS -V
#PBS -A mp999

cd $PBS_O_WORKDIR

echo "Starting at" `date`

aprun -n 4096 -N 8 ./code.x
```

PBS Directives vs. aprun options

Jaguar & Kraken

aprun –n value must be consistent with
#PBS –l size

Size must be a multiple of 12

You will always get size/12 nodes

```
#PBS -l walltime=01:00:00
#PBS -l size=3072
###NOTE: 256 nodes
#PBS -N BigJob
#PBS -V
#PBS -A mp999


cd $PBS_O_WORKDIR

echo "Starting at" `date`

aprun -n 3072  ./code.x
```

Interactive Jobs

You can run interactive parallel jobs. It may not make semantic sense, but you can think of this as an interactive batch job: PBS/ Torque. Moab, ALPS all participate.

```
% qsub -I -l mppwidth=24 -l walltime=30:00 Hopper
% qsub -I -l size=24 -A acct -lwalltime=30:00
                                        Jaguar, Kraken
… wait for prompt …
```

When your prompt returns, you are on a service node, but you have compute nodes reserved for you so you can use `aprun` at the command line

```
nid04100% cd $PBS_O_WORKDIR
nid04100% aprun -n 24 ./mycode.x
```

aprun will fail if you don't first use `qsub -I` to reserve compute nodes.

Job Notes

- Work out of $SCRATCH, /tmp/work/$USER, or /lustre/scratch/$USER.
- The job script itself executes on a service (MOM) node.
- All commands and serial programs (including hsi) therefore run on a shared node running a full version of Linux.
- You must use aprun to run anything on the compute nodes.

OpenMP

Each MPI task (instance) can create multiple OpenMP threads.

In most cases you will want 1 OpenMP thread per core.

You will need to use the proper PBS/Torque directives and aprun options.

Examples follow.

OpenMP on Hopper

Hopper

Run using 4 MPI tasks per node and 2 OpenMP threads per task. Run 2 MPI tasks per socket.

```
#PBS -l walltime=00:30:00
#PBS -l mppwidth=1024
#PBS -l mppnppn=4
#PBS -l mppdepth=2
#PBS -q debug
#PBS -N BigOpenMPJob
#PBS -V

cd $PBS_O_WORKDIR
setenv OMP_NUM_THREADS 2
aprun -n 1024 -N 4 -S 2 -d 2 ./OMPcode.x
```

OpenMP on Jaguar/Kraken

Jaguar & Kraken

Run using 6 MPI tasks per node and 2 OpenMP threads per task. Run 3 MPI tasks per socket.

```
#PBS -l walltime=00:30:00
#PBS -l size=1008
#PBS -q debug
#PBS -N BigOpenMPJob
#PBS -V

cd $PBS_O_WORKDIR

setenv OMP_NUM_THREADS 2

aprun -n 1008 -N 6 -S 3 -d 2 ./OMPcode.x
```

Outline

- XT5 Overview
- Creating and Submitting a Batch Job
- How a Job Is Launched
- Monitoring Your Job
- Queues and Policies

Monitoring Jobs

- **Monitoring commands – each shows something different**
  - **showq – moab**
  - **qstat – torque**
  - **showstart – moab**
  - **checkjob – moab**
  - **apstat – ALPS**
  - **xtshowcabs/xtnodestat – Cray**
  - **qs – NERSC's concatenation**

showq (moab)

```
                  active jobs------------------------
JOBID              USERNAME      STATE PROCS    REMAINING            STARTTIME

 249696                ptr    Running     2     00:20:20  Tue Sep 18 14:21:13
249678                 puj    Running    32     00:24:43  Tue Sep 18 13:55:36

                  eligible jobs----------------------
JOBID              USERNAME      STATE PROCS     WCLIMIT            QUEUETIME

249423             toussain       Idle  8192     3:00:00  Tue Sep 18 05:21:30
249424             toussain       Idle  8192     3:00:00  Tue Sep 18 05:21:35

                  blocked jobs-----------------------
JOBID              USERNAME      STATE PROCS     WCLIMIT            QUEUETIME

248263              streuer       Hold  4096    12:00:00  Sat Sep 15 10:27:06
248265              streuer       Hold  2048    12:00:00  Sat Sep 15 10:27:06
```

qstat –a (torque)

```
                                                                  Req'd  Req'd   Elap
Job ID                  Username Queue    Jobname    SessID NDS   TSK Memory Time  S Time
----------------------- -------- -------- ---------- ------ ----- --- ------ ----- - -----
248262.nid00003         streuer  reg_2048 td4         17483   --    --    --  12:00 R 10:03
 248263.nid00003         streuer  reg_2048 td4           --    --    --    --  12:00 H   --
 248265.nid00003         streuer  reg_1024 td1024        --    --    --    --  12:00 H   --
 248266.nid00003         streuer  reg_1024 td1024        --    --    --    --  12:00 H   --
248806.nid00003         toussain reg_2048 gen1        773    --    --    --  05:00 R 03:15
 248826.nid00003         u4146    reg_512  B20_GE2_k1    --    --    --    --  12:00 Q   --
 248845.nid00003         toussain reg_2048 spec1         --    --    --    --  05:00 Q   --
 248846.nid00003         toussain reg_2048 gen1          --    --    --    --  05:00 Q   --
 248898.nid00003         u4146    reg_1024 BW_GE2_36k    --    --    --    --  12:00 Q   --
 248908.nid00003         u4146    reg_2048 VS2_GE2_k1    --    --    --    --  06:00 Q   --
 248913.nid00003         lijewski reg_1024 doit          --    --    --    --  06:00 Q   --
248929.nid00003         aja      reg_512  GT1024V4R   21124    --    --    --  12:00 R 08:51
 248931.nid00003         aja      reg_512  GT1024IR      --    --    --    --  12:00 Q   --
```

Blank

Random order

Showstart (moab)

```
nid04100% showstart 249722.nid00003
job 249722 requires 8192 procs for 2:00:00

Estimated Rsv based start in                 4:46:10 on Tue Sep 18 20:13:05
Estimated Rsv based completion in            6:46:10 on Tue Sep 18 22:13:05

Best Partition: hopper
```

May not be very useful, assumes that you are “top dog,” i.e., “next”?

apstat

```
Compute node summary
   arch config     up    use   held  avail   down
     XT    664    663    468      9    186      1

No pending applications are present

Total placed applications: 30
Placed  Apid ResId     User   PEs Nodes    Age   State Command
      148083    63  groucho   128    16  66h57m   run   cp_dis7_again
      150341   301   freddy   512    64  12h45m   run   parsec.mpi
      150422  1051     jyma     1     1   5h48m   run   tri.x
      150433  1134     ynwu    32     4   5h35m   run   vasp
      150441  1173  kokomoj    64     8   5h23m   run   pw.x
      150440  1174  freddie    64     8   5h23m   run   pw.x
      150445  1175  wanda23    64     8   5h16m   run   pw.x
      150448  1184    jimmy    64     8   5h14m   run   vasp.5.mpi
      150455  1206  afrankz   256    32   5h09m   run   namd2
      150456  1209  tpaudel   112    14   4h59m   run   vasp5
```

xtnodestat or xtshowcabs

```
     C0-0     C1-0     C2-0     C3-0     C4-0     C5-0     C6-0     C7-0
  n3 mSemmmmS SqSqqq*q wwwSwww* AAAAAAqq wwwwwwww kkkkSmAA wwkkkSkk kkkkkkSk
  n2 k kkkkk   - --kmm kkk k--c ----k--- cccccccc ---c c-- ccccc -- ------ -
  n1 e eeeee   k kkkkk eee ---- -------- -------- vvvv v-k --ccc cc cccvvv v
c2n0 -S-----S S-S---*e ---Skkk* eeeeeee- kk------ --eeSeee -----S-- ------S-
  n3 --S--j*j ---S---* *jjjSjjc -------- cccj---c -----S-- ccccccSc cccc---S
  n2 ii iiiii cc- ---- iiii iiv -------- vvvvvvvv ----- -- vvvvvv v vvvvvvv
  n1 ee eeeee ppp pppi e--- --e ccpccppp eeeeeeek pppcc cc kkkkkk k ccffppp
c1n0 ccSccc*c -kkS---* *cccSccc cc----ce cXcccsss cccccScc s----cSc cccccccS
  n3 SS--*--- ScccSccc S----S-- --cccccc -------- ------S- -------S ---S----
  n2   ceeefh  ooo o--  hhhk kk cccccekk ekkkyycc cccccc c ccccckk  kkk yyyc
  n1   bddbbb  eee eec  bdoo oo eeeeeeee yyzzzzze cccccc c eCEEEEE  exe eeyn
c0n0 SSaA*a-g SnbbSbfk SrstuSuu fddxnnnn u----ttt fdfdfzSB dCDBBuuS uuBSBxxx
    s01234567 01234567 01234567 01234567 01234567 01234567 01234567 01234567

Legend:
   nonexistent node                  S  service node
;  free interactive compute CNL      -  free batch compute node CNL
A  allocated, but idle compute node  ?  suspect compute node
X  down compute node                 Y  down or admindown service node
Z  admindown compute node            R  node is routing
*  system dedicated node (DVS)

Available compute nodes:          0 interactive,        202 batch

Job ID     User       Size   Age              command line
--- ------ -------- -----  ---------------  ---------------------------------
a   150577 ynwu     2      0h15m            vasp
b   150445 peterpp   8      5h22m            pw.x
c   150078 mikre     128    22h35m           pmemd
d   150448 luckisg   8      5h29m            pw.x
e   150341 jxhan    64     12h51m           parsec.mpi
f   150458 gfwu     8      5h04m            mvasp4631
```

qs (NERSC)

Jobs shown in run order.

```
nid04108% qs
  JOBID ST      USER          NAME  SIZE       REQ      USED              SUBMIT
 250029  R       abc  md_e412.su    32  01:00:00  00:37:49   Sep 18 22:00:56
 249722  R       cmc  MADmap_all  8192  02:00:00  00:37:48   Sep 18 15:14:55
 249477  R        de       spec1  8192  03:00:00  00:37:48   Sep 18 09:11:22
 249485  R       dks  test.scrip   144  12:00:00  00:36:57   Sep 18 09:21:03
 249666  R       dks  test.scrip   192  12:00:00  00:36:58   Sep 18 13:42:35
 248898  R       llp  BW_GE2_36k  2592  12:00:00  00:36:26   Sep 17 03:30:28
 248845  Q  bunnysl1       spec1  4096  05:00:00         -   Sep 16 20:21:15
 248846  Q  tunnel2        gen1  4096  05:00:00         -   Sep 16 20:21:21
 248908  Q     davey  VS2_GE2_k1  6144  06:00:00         -   Sep 17 07:12:53
 248913  Q  goliath3        doit  2048  06:00:00         -   Sep 17 07:52:13
 248931  Q       aja    GT1024IR  1024  12:00:00         -   Sep 17 09:29:28
```

NERSC web queue display:

https://www.nersc.gov/nusers/status/queues/hopper/

Outline

- XT5 Overview
- Creating and Submitting a Batch Job
- How a Job Is Launched
- Monitoring Your Job
- Queues and Policies

Batch Queues & Policies

See web pages.

Hopper

http://www.nersc.gov/nusers/systems/hopper/running_jobs/queues.php

Kraken

http://www.nics.tennessee.edu/computing-resources/kraken/running-jobs/queues

Jaguar

http://www.nccs.gov/computing-resources/jaguar/running-jobs/scheduling-policy-xt5/