Running Parallel Jobs

Cray XE6 Workshop
February 7, 2011

David Turner
NERSC User Services Group

U.S. DEPARTMENT OF ENERGY | Office of Science

NERSC National Energy Research Scientific Computing Center

BERKELEY LAB Lawrence Berkeley National Laboratory

Hopper Compute Nodes

- **6,384 nodes (153,216 cores)**
  - 6000 nodes have 32 GB; 384 have 64 GB
- **Small, fast Linux OS**
  - Limited number of system calls and Linux commands
  - No shared objects by default
    - Can support ".so" files with appropriate environment variable settings
- **Smallest allocatable unit**
  - Not shared

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB Lawrence Berkeley National Laboratory

Hopper Login Nodes

- **8 nodes (128 cores)**
  - 4 quad-core AMD 2.4 GHz processers
  - 128 GB
  - Full Linux OS
- **Arbitrary placement upon login**
  - Load balanced via number of connections
- **Edit, compile, submit**
  - No MPI
- **Shared among many users**
  - CPU and memory limits

Hopper MOM Nodes

- **24 nodes**
  - 4 quad-core AMD 2.4 GHz processers
  - 32 GB
- **Launch and manage parallel applications on compute nodes**
- **Commands in batch script are executed on MOM nodes**
- **No user (ssh) logins**

File Systems

- **$HOME**
  - Tuned for small files
- **$SCRATCH**
  - Tuned for large streaming I/O
- **$PROJECT**
  - Sharing between people/systems
  - By request only

Running on Login Nodes

```
% cc hello.c
% ./a.out
Hello, world!
```

- Login nodes are not intended for computation!
- No MPI!

How to Access Compute Nodes

- **Requires two components**
  - Batch System
    - Based on PBS
      - Moab scheduler
      - Torque resource manager
    - qsub command
    - Many monitoring methods
      - qs, qstat, showq, NERSC website, …
  - Application Launcher
    - aprun command
      - Similar to mpirun/mpiexec

Basic qsub Usage

```
% cat myjob.pbs
#PBS -l walltime=00:10:00
#PBS -l mppwidth=48
#PBS -q debug
cd $PBS_O_WORKDIR
aprun -n 48 ./a.out
% qsub myjob.pbs
140979.sdb
```

Batch Queues

Submit Queue	Execute Queue	Max Nodes	Max Cores	Max Walltime
interactive	interactive	256	6,144	30 mins
debug	debug	512	12,288	30 mins
regular	reg_short	512	12,288	6 hrs
	reg_small	512	12,288	12 hrs
	reg_med	4,096	98,304	12 hrs
	reg_big	6,384	153,216	12 hrs
low	low	512	12,288	6 hrs

Batch Options

- **-l walltime=*hh:mm:ss***
- **-l mppwidth=*num_cores***
  - Determines number of nodes to allocate; should be a multiple of 24
- **-l mpplabels=bigmem**
  - Will probably have to wait for bigmem nodes to become available
- **-q *queue_name***

Batch Options

- **-N *job_name***
- **-o *output_file***
- **-e *error_file***
- **-j oe**
  - Join output and error files

Batch Options

- **-V**
  - Propagate environment to batch job
- **-A *repo_name***
  - Specify non-default repository
- **-m *[a|b|e|n]***
  - Email notification
  - abort/begin/end/never

Running Interactively

```
% qsub -I -V
-l walltime=00:10:00
-l mppwidth=48 -q interactive
qsub: waiting for job 140979.sdb
to start
qsub: job 140979.sdb ready
% cd $PBS_O_WORKDIR
% aprun -n 48 ./a.out
```

Packed vs Unpacked

- **Packed**
  - User process on every core of each node
  - One node might have unused cores
  - Each process can safely access ~1.25 GB
- **Unpacked**
  - Increase per-process available memory
  - Allow multi-threaded processes

Packed

```
#PBS -l mppwidth=1024
aprun –n 1024 ./a.out
```

- **Requires 43 nodes**
  - 42 nodes with 24 processes
  - 1 node with 16 processes
    - 8 cores unused
  - Could have specified mppwidth=1032

Unpacked

```
#PBS -l mppwidth=2048
aprun -n 1024 -N 12 ./a.out
```

- **Requires 86 nodes**
  - 85 nodes with 12 processes
  - 1 node with 4 processes
    - 20 cores unused
  - Could have specified mppwidth=2064
  - Each process can safely access ~2.5 GB

Manipulating Batch Jobs

- qsub *job_script*
- qdel *job_id*
- qhold *job_id*
- qrls *job_id*
- qalter *new_options job_id*
- qmove *new_queue job_id*

Monitoring Batch Jobs

- **qstat –a [-u *username*]**
  - All jobs, in submit order
- **qstat –f *job_id***
  - Full report, many details
- **showq**
  - All jobs, in priority order
- **qs [-w] [-u *username*]**
  - NERSC wrapper, priority order
- **apstat, showstart, checkjob, xtnodestat**

Hands-On

/project/projectdirs/training/XE6-feb-2011/RunningParallel

```
jacobi_mpi.f90
jacobi.pbs
indata

mmsyst.f
mmsyst.pbs
```