New York University
High Performance Computing

High Performance Computing
Information Techonology Services
New York University
hpc@nyu.edu

September 20, 2011

Outline of Topics

NYU ITS HPC Service

- High Performance Computing, Information Techonology Services, (HPC/ITS)
- HPC service started from 2005
- HPC resources: clusters, storage, software with site licenses and open source
- HPC resources are open to NYU faculty, staff, faculty-sponsored students, and for class instruction
- HPC accounts application and renewal
- `https://wikis.nyu.edu/display/NYUHPC/High+Performance+Computing+at+NYU`
- NYU HPC maintains three main clusters: USQ, Bowery, Cardiac
- NVIDIA GPU nodes

NYU HPC Cluster: USQ

Union Square: usq.es.its.nyu.edu

- 2 login nodes, 140 compute nodes, 8 CPU cores per node
- Memory: compute-0-0 to 115 have 16GB, 116 to 139 have 32GB
- Intel(R) Xeon(R) CPU @ 2.33GHz
- Mainly for serial jobs
- 124 compute nodes are online now
- In production from 2007, will be retired in 2012 summer after more than 4 years service

NYU HPC Cluster: Bowery

Bowery: bowery.es.its.nyu.edu
Owned by ITS and Center for Atmosphere Ocean Science (CAOS)

- 4 login nodes
- 160 compute nodes with Intel(R) Xeon(R) CPU @ 2.67GHz
  - 64 nodes, 8 CPU cores, 24GB memory
  - 72 nodes, 12 CPU cores, 24GB memory
  - 8 nodes, 12 CPU cores, 48GB memory
  - 16 nodes, 12 CPU cores, 96GB memory
- 1 node with Intel(R) Xeon(R) CPU @ 2.27GHz, 16 CPU cores and 256GB memory
- First setup with 64 nodes in 2009, expanded to 161 nodes in 2010
- Bowery is mainly for multiple-node MPI parallel jobs and big memory serial jobs

NYU HPC Cluster: Cardiac

Cardiac: cardiac1.es.its.nyu.edu
Owned by ITS and Prof. Charles S. Peskin in CIMS

- 1 login node, 79 compute nodes
- 16 CPU cores and 32GB memory on each compute node
- Quad-core AMD Opteron(tm) Processor 8356
- Cardiac is for parallel and serial jobs

NYU HPC Cluster: CUDA

- NVIDIA GTX 285, setup in 2009 spring
- 4 nodes with NVIDIA M2070 (will be in production soon ...)
  - Peak double precision floating point performance: 515 GFlops
  - Peak single precision floating point performance: 1030 GFlops
  - Memory (GDDR5): 6 GB
  - 448 CUDA cores

Login to HPC Clusters

Connect to NYU HPC clusters via SSH (Secure Shell)
SSH client + X server

- Windows: PuTTY (Free Telnet/SSH Client) + Xming (PC X server)
- Linux/Unix/Mac OS: Terminal + X11 client utilities

Login steps

- From your desktop to NYU HPC bastion host: `hpc.nyu.edu`

  ```
  ssh sw77@hpc.nyu.edu
  ```

- From bastion host to HPC clusters
  - to USQ: `ssh usq`
  - to Bowery: `ssh bowery`
  - to Cardiac: `ssh cardiac1`

X11 forwarding with SSH: enable `-X` flag for `ssh`

```
ssh -X sw77@hpc.nyu.edu
ssh -X usq
```

Storage allocations

There are 4 file systems on each HPC clusters

- home: `/home/sw77`
  - quota = 5GB
  - local to each cluster, accessed from login nodes and compute nodes
  - space to save source code, scripts, libraries, executable files ...
  - backup
- scratch: `/scratch/sw77`
  - quota = 5 TB, data will be forced to clean up when the free space is small
  - shared file system, accessed from the login nodes and compute nodes on all the 3 clusters
  - space for job running, data analysis, scratch files, ...
  - no backup
- local scratch on the compute nodes: `/state/partition1/sw77`
  - local to each compute node, save scratch and temporary files, mainly for quantum chemistry applications
- archive: `/archive/sw77`
  - quota = 2TB
  - shared file system, accessed from the login nodes on all the 3 clusters
  - space for data storage only
  - backup

Copy Data from/to HPC Clusters

- Use SCP to transfer files
  - local desktop $\rightleftarrows$ bastion host $\rightleftarrows$ HPC login node
  - HPC compute node $\rightarrow$ bastion host $\rightleftarrows$ local desktop
  - best and easist way
    - **HPC login node or compute node $\rightarrow$ local desktop**
- scp usage (for Linux or Mac OS)

  `scp [[user@]from-host:]source-file [[user@]to-host:][destination-file]`
- For Windows users, use WinSCP from local desktop to bastion host
- Examples:
  - On desktop: `scp -rp Amber11.pdf sw77@hpc.nyu.edu:~/.`
  - On bastion host: `scp -rp Amber11.pdf usq:~/.`
  - On USQ login node or compute node:
    - `scp -rp hpc.nyu.edu:~/Amber11.pdf .`
    - `scp -rp Amber11.pdf wangsl@wangmac.es.its.nyu.edu:~/.`
- Do not keep heavy data on the bastion host
- **SCP through SSH tunneling**

  direct copy data from local desktop to HPC clusters

  `https://wikis.nyu.edu/display/NYUHPC/SCP+through+SSH+Tunneling`

Queue Settings

`https://wikis.nyu.edu/display/NYUHPC/Queues`

- Job scheduler: Moab/TORQUE
- interactive: 4 hours, 2 nodes maximum
- p12: parallel jobs, 12 hours maximum, 2 nodes minimum
- p48: parallel jobs, 48 hours maximum, 2 nodes minimum
- ser2: serial jobs, 1 node ($\leq$ 8 or 16 CPU cores), 48 hours
- serlong: serial jobs, 1 node ($\leq$ 8 or 16 CPU cores), 96 hours
- bigmem: for jobs with more memories, serial or parallel jobs

Queue settings for general users on HPC clusters

- USQ: interactive, ser2, serlong, p12, bigmem (14GB $\leq$ mem $\leq$ 30GB)
- Bowery: interactive, p12, p48, bigmem (22GB $\leq$ mem $\leq$ 254GB)
- Cardiac: interactive, ser2, serlong, p12, p48
- Please always declare the proper wall time in order to use the proper queue

Runing Jobs with Portable Batch System (PBS)

- Login nodes are for login, text editor, file transfer, simple cron jobs in the background, ...
- Compute nodes are for job running, source code compiling, debugging, ...
- Checkout 1 or 2 compute nodes with all the 8 CPU cores for 4 hours from interactive queue

```
qsub -I -q interactive -l nodes=1:ppn=8,walltime=04:00:00
qsub -I -q interactive -l nodes=2:ppn=8,walltime=04:00:00
```

- Interactive queue wirh X11 forwarding, turn on flag `-X`

```
qsub -X -I -q interactive -l nodes=1:ppn=8,walltime=04:00:00
```

- Interactive jobs for more than 4 hours

```
qsub -X -I -q serlong -l nodes=1:ppn=8,walltime=96:00:00
```

Never try to run heavy jobs on the login nodes

Available Software

- Third party software installed into the path `/share/apps` accessed from both login nodes and compute nodes
- Show available software `module avail`

```
[yz22@login-0-0 ~]$ module ava
---------------------------------------- /share/apps/modules/modulefiles ----------------------------------------
R/intel/2.13.0              gcc/4.5.2                   matlab/R2009b               openmpi/intel/1.2.8
R/intel/2.9.2               git/gnu/1.7.2.3             matlab/R2010b               openmpi/intel/1.3.3
amber/amber10               gnuplot/gnu/4.2.6           matlab/R2011a               openmpi/intel/1.4.3
amber10/intel-mvapich       gnuplot/gnu/4.4.2           mesa/gnu/7.6                openssl/gnu/0.9.8o
amber11/intel-mvapich       grace/intel/5.1.22          migrate-n/intel/3.0.8       perl-module/5.8.8
apbs/intel/1.2.1            gsl/gnu/1.13                mkl/11.1.046                python/2.6.4
arlequin/3.5.1.2            gsl/intel/1.12              mltomo/mvapich/intel/1.0    qt/gnu/3.3.8b
ati-stream-sdk/2.2          gsl/intel/1.13              molden/gnu/4.7              qt/gnu/4.7.1
autodocksuite/intel/4.2.1   hdf/intel/1.8.4/parallel    mpiexec/gnu/0.84            root/intel/5.24.00
bayescan/gnu/2.01           hdf/intel/1.8.4/serial      mpiexec/intel/0.84          root/intel/5.27.04
boost/intel/1.44.0/openmpi  hdf/intel/1.8.7/serial      mpiexec84/mpiexec84         shrimp/intel/2.2.0
boost/intel/1.44.0/serial   ibm-java/1.6.0              mvapich/gnu/1.1.0           stata/11
charmm/intel/c35b5/mvapich  intel/11.1.046              mvapich/intel/1.1.0         tcl/gnu/8.5.8
charmm/intel/c35b5/serial   intel-c/cce/10.0.023        mvapich/intel/1.1rc1        tinker/intel/4.2
cmake/gnu/2.8.1             intel-c/cce/11.1.046        namd/intel/2010-06-29       tinker/intel/5.0
elmerfem/intel/svn5119      intel-fortran/fce/10.0.023  ncbi/intel/2.2.21           totalview/8.8.0-2
expat/intel/2.0.1           intel-fortran/fce/11.1.046  ncl/gnu/5.2.0               valgrind/gnu/3.6.0
fftw/gnu/2.1.5              jdk/1.6.0_24                ncl/gnu/6.0.0               vapor/gnu/1.5.2
fftw/intel/2.1.5            ldhat/intel/2.1             ncview/intel/1.93g          vmd/1.8.7
fftw/intel/3.2.2            maple/15                    netcdf/intel/3.6.3          vmd/1.9
gaussian/intel/G03-D01      maq/intel/0.7.1             netcdf/intel/4.1.1          xmipp/openmpi/intel/2.4
gaussian/intel/G03-E01      mathematica/7.0             netcdf/intel/4.1.2          xxdiff/gnu/3.2
gaussian/intel/G09-B01      mathematica/8.0             neuron/intel/7.1
gaussview/5.0.9             matlab/R2009a               openmpi/gnu/1.2.8
```

Available Software

- load module `module load matlab/R2011a`
- unload module `module unload matlab/R2011a`
- Remove all the modules `module purge`

```
[yz22@login-0-0 ~]$ which matlab
matlab: Command not found.
[yz22@login-0-0 ~]$ module load matlab/R2011a
[yz22@login-0-0 ~]$ which matlab
/share/apps/matlab/R2011a/bin/matlab
[yz22@login-0-0 ~]$ module unload matlab/R2011a
[yz22@login-0-0 ~]$ which matlab
matlab: Command not found.
[yz22@login-0-0 ~]$ module load matlab/R2009b
[yz22@login-0-0 ~]$ which matlab
/share/apps/matlab/R2009b/bin/matlab
[yz22@login-0-0 ~]$ module purge
[yz22@login-0-0 ~]$ which matlab
matlab: Command not found.
```

PBS Script Example

PBS script = PBS directives + shell script

```
#!/bin/sh

#PBS -l nodes=1:ppn=4,walltime=05:00:00
#PBS -N matlab-test
#PBS -M sw77@nyu.edu
#PBS -m abe

source /etc/profile.d/env-modules.sh
module load matlab/R2011a

cd /scratch/sw77/zzz

cat<<EOF | matlab -nodisplay -multipleCompThreads > run.log 2>&1
warning off MATLAB:maxNumCompThreads:Deprecated
maxNumCompThreads(4);

n = 12000; pi = 4.0*atan(1.0);
t = clock;
A = sin(rand([n,n], 'double')*pi);
B = sin(2.0*rand([n,n], 'double')*pi);
A*B - B*A;
etime(clock, t)
exit
EOF
```

To submit job on the login nodes: `qsub run-matlab.pbs`

Submit and Monitor Jobs

- Submit jobs
  - `qsub PBS_Script_Name`
  - Job array: `qsub PBS_Script_Name -t 0-299%5`
  - Job dependency:

    ```
    qsub -W depend=afterok:42785.crunch.local job2.pbs
    qsub -W depend=afterany:42785.crunch.local job2.pbs
    ```
- Monitor the jobs
  - `showq` or `showq -u sw77`
  - `qstat` or `qstat -u sw77`
  - `pbstop`
- Kill jobs
  - `qdel JobID` or `qdel all`
  - `canceljob JobID` or `canceljob all`
- Reference
  - `https://wikis.nyu.edu/display/NYUHPC/High+Performance+Computing+at+NYU`
  - `http://www.clusterresources.com/products/mwm/docs/commands/mjobctl.shtml`
  - `hpc@nyu.edu`

Matlab Parallel Computations

- NYU has site license for Matlab and most of the toolboxes
- Matlab is available on NYU HPC clusters
- Paralle computing with Matlab
  - Built-in Matlab functions with multithreaded computation to use multiple CPU cores on 1 compute node
  - Matlab Executable (MEX) functions compiled from OpenMP code with C/C++/Fortran
  - Parallel Computing Toolbox (PCT) to use maximum 8 workers on the same node
  - Parallel Computing Toolbox (PCT) with GPU
  - Distributed Computing Server (DCS) to run Matlab jobs with more than 8 workers or accross multiple compute nodes
  - NYU ony has site license for PCT, not for DCS

Matlab Multithreads

- Matlab will turn on multithreads automatically by default, it can be disabled by the flag `-singleCompThread`
- Multithreads has been turned off by default on NYU HPC clusters, it can be enabled by the flag `-multipleCompThreads`
- Control the multithreads number by the deprecated function `maxNumCompThreads`
- Setting the maximum number of computational threads using `maxNumCompThreads` does not propagate to your next MATLAB session

Matlab Parallel Computing Toolbox

PBS scripts to use Matlab PCT

Multiple Matlab parallel jobs with PCT running simultaneously for one user

```
#!/bin/sh

#PBS -V
#PBS -N par-matlab
#PBS -l nodes=1:ppn=8,walltime=01:10:00
#PBS -q ser2

source /etc/profile.d/env-modules.sh
module load matlab/R2011a

cd /home/sw77/MatLab/pi-parallel

export DATA_LOCATION=$(mktemp -d "/state/partition1/MATLAB-data-XXXXXXXXXX")
export NTHREADS=$(cat $PBS_NODEFILE | wc -l)

matlab -nodisplay < myrun.m > run.log 2>&1

exit 0
```

Matlab Parallel Computing Toolbox

$$\int_0^1 dx \, \frac{4}{1+x^2} = \pi$$

```
function f = myf(x)
f = 4.0/(1.0 + x*x);
return

function pi = mypi_par(n)

a = 0.0; b = 1.0; dx = (b-a)/(n-1);

s = 0.0;
parfor i = 1 : n
  x = (i-1)*dx + a;
  fx = myf(x);
  s = s + fx;
end

s = (s - 0.5*(myf(a) + myf(b)))*dx;
pi = s;
fprintf(1, ' Pi = %.16f\n', pi);
return
```

```
t_begin = clock;

data_location = getenv('DATA_LOCATION');
nthreads = getenv('NTHREADS');

scheduler = findResource('scheduler', 'type', 'local')
scheduler.DataLocation = data_location

matlabpool('open', 'local', nthreads)

n = 524288*16*8;

for i = 1 : 20
  fprintf(1, '\n ***** %d *****\n', i);
  t = clock;
  mypi_par(n);
  time_elapsed = etime(clock, t);
  fprintf(1, ' Elasped time: %.2f\n', time_elapsed);
end

matlabpool close

exit
```

Matlab Parallel Computing Toolbox

```
scheduler =

Local Scheduler Information
===========================

                      Type : local
              ClusterOsType : unix
                ClusterSize : 8
               DataLocation : /home/sw77/.matlab/local_scheduler_data/R2011a
       HasSharedFilesystem : true

- Assigned Jobs

           Number Pending  : 0
           Number Queued   : 0
           Number Running  : 0
           Number Finished : 0

- Local Specific Properties

         ClusterMatlabRoot : /share/apps/matlab/R2011a
>>
scheduler =

Local Scheduler Information
===========================

                      Type : local
              ClusterOsType : unix
                ClusterSize : 8
               DataLocation : /state/partition1/MATLAB-data-ftFDz11609
       HasSharedFilesystem : true

- Assigned Jobs

           Number Pending  : 0
           Number Queued   : 0
           Number Running  : 0
```

Thanks for your attention!

Any question please send email to
hpc@nyu.edu