HPC Resources at Lehigh

Steve Anthony
March 22, 2012

HPC at Lehigh: Resources

What's Available?

Service Level Basic	Service Level E-1	Service Level E-2
Leaf and Condor Pool	Altair	Trits, Cuda0, Inferno, and Corona
Default	$0 Request*	$450/yr Request*

* https://cf.lehigh.edu/accounts/HPCAccountRequests/

How to Access Resources

SSH is the protocol used to access the HPC resources at Lehigh.

- OS X: Applications->Utilities->Terminal and type ssh LehighID@hostname
- Windows: SSH Client available at Public Sites and www.lehigh.edu/software.

GUI application access requires X-forwarding.

- OS X: In the Terminal, type ssh -X *hostname* (X11 client needed).
- Windows: Xming and X-Win32 available at lehigh.edu/software.
  - http://www.lehigh.edu/computing/hpc/services/ssh_xming.html

Off-campus access requires VPN or SSH gateway.

- VPN information available at lehigh.edu/vpn.
- To use SSH gateway connect to ssh.lehigh.edu then connect to resource needed using ssh hostname.

Filesystem Map

- /ahome -- Altair home directory.
- /bhome -- Blaze (Inferno) home directory.
- /vhome -- Vega old home directories; links to other (current) home directories.
- /zhome -- NAS2 user directories (available by request).
- /zhome/Apps -- NAS2 software installations common to all systems.
- /Projects -- NAS1 user/group projects.
- /usr/local -- local software installations.

Level Basic: LEAF

System Statistics

- 39 Nodes, 4 Intel 686 CPUs (32-bit), 12GB RAM per node. Hostname is leaf.cc.lehigh.edu.

Lehigh Application Farm

- Lots of software available, including COMSOL, ABAQUS, Matlab. Software can be run directly or scheduled through Condor.
- Full list at www.lehigh.edu/software

Better for computation than visualization, but we're working to improve the latter!

AFS Quota Woes

AFS used for home on LEAF:

- 100MB (faculty/graduate students quota)
- 50MB (undergraduates quota)

Problem: COMSOL generates 200MB+ configuration data at application start-up!

Three options: /zhome, /Projects, and /tmp.

- /zhome -- File-locking. Per-user storage.
- /Projects -- No File-locking. User/group storage.
- /tmp -- Temporary Storage. No request needed.

Level Basic: Condor Pool

Composed of most LTS Public Sites.
Allows users to make use of idle systems.
High Throughput Computing resource.

```
Name                    OpSys          Arch    State     Activity LoadAv Mem
ActivityTime
slot9@leaf3.cc.leh        LINUX     INTEL  Unclaimed Idle     0.220   995  0+00:05:04
slot9@leaf30.cc.le      LINUX     INTEL  Unclaimed Idle     0.210   995  0+00:00:04
slot9@leaf31.cc.le      LINUX     INTEL  Owner     Idle     0.450   995  0+00:05:04
slot9@leaf32.cc.le      LINUX     INTEL  Unclaimed Idle     0.240   995  0+00:00:04
slot7@ph317c.cc.le      LINUX     X86_64 Owner     Idle     1.000   998  1+05:15:12
slot8@ph317c.cc.le      LINUX     X86_64 Owner     Idle     1.400   998  1+05:15:05
slot1@PS-DROWN10O.       WINNT61   INTEL  Claimed    Busy     1.030   989  0+01:52:52
slot2@PS-DROWN10O.       WINNT61   INTEL  Claimed    Busy     1.030   989  0+01:51:49
```

Using Condor

Access and submit jobs through LEAF.

- Setup on LEAF by running:
  - . /Projects/condor/32/condor/condor.sh
- View pool availabilities by running:
  - condor_status
- Compiling code for Condor:
  - Condor can run binaries or code compiled without modifications in the vanilla universe.
  - If you want features like checkpointing, must recompile using: condor_compile gcc test.c -o test
  - If using Makefiles,must start each call to g++, gcc, f77, gfortran etc. with condor_compile.

Condor Sample & Commands

```
########################
# This is a comment.
# Submit file for a simple program located at
# /home/username/some/path/my_file
########################

universe = vanilla
notify_user = username@lehigh.edu
notification = Always
getenv = TRUE
initialdir = /home/username/dir1

Executable = /home/username/some/path/my_file
Output = foo.out.$(PROCESS)
log = foo.log.$(CLUSTER)
error = foo.err.$(PROCESS)
arguments = arg01 arg02

should_transfer_files = YES
transfer_input_files =
when_to_transfer_output = ON_EXIT_OR_EVICT

queue 10
```

condor_submit file.submit -- submit a job to Condor using the option in file.submit.

condor_status -- check the status of hosts in the condor pool.

condor_q -global sma310 -- show the status of all jobs submitted by sma310.

condor_q -global -l 701090.0 -- show the status of job 701090.0.

condor_rm 701090.0 -- remove job 701090.0 from the queue.

condor_rm sma310 -- remove all jobs submitted by sma310 from the queue.

Level E-1: Altair

System Statistics

- 8 quad core 2.3 GHz Intel Xeon CPUs (64-bit), 128GB RAM, 1400GB home, 100MB user quotas.
- Hostname is altair.cc.lehigh.edu.

Scheduling

- Condor scheduling encouraged, not currently required.

Level E-2: The Trits, Cuda0, and Capella

Trit1, Trit2, Trit3 (Condor)

2x 2.93GHz Intel Xeon X5570 (Nehalem) quad core 64-bit, 48GB RAM, 500GB disk. Trit1 is batch only (Condor). Interactive login permitted on Trit2 and Trit3. Hostnames: trit1.cc.lehigh.edu trit2.cc.lehigh.edu, trit3.cc.lehigh.edu.

Cuda0 (Unscheduled)

2x nVidia Tesla C2050 GPUs. Each has 3GB of GDDR5 memory and support single and double precision floating point operations.

2x nVidia Tesla C2070 GPUs. Each has 4GB of GDDR5 memory and support single and double precision floating point operations.

Hostname: cuda0.cc.lehigh.edu

Capella (Condor)

4x Opteron 8384 (Shanghai) quad core 64-bit CPU, 64GB RAM, 2x 146GB SAS 15k RPM drives. Hostname capella.cc.lehigh.edu

Level E-2: Inferno (Blaze)

System Statistics

40 Nodes, Dual quad-core Xeon 1.8GHz, 16GB RAM per node. Hostname is inferno.cc.lehigh.edu.

Condor submission

- Similar to Condor Pool, but can use MPI with parallel universe.
- Users need to setup SSH-keys after opening account to use with: setup_mpi_ssh

```
universe = parallel
notify_user = user-name@lehigh.edu
notification = Error
getenv = TRUE

# this is where condor output and log
appear
initialdir = /home/user-name/some/path

Executable = /usr/local/bin/mp1script
machine_count = 4
Output = foo.out.$(NODE)
log = foo.log.$(CLUSTER)
error = foo.err.$(NODE)

# name of the MPI-binary
arguments = /home/user-
name/path/to/mpi_foo
     arg01 arg02

should_transfer_files = YES
transfer_input_files =
WhenToTransferOutput = ON_EXIT_OR_EVICT

queue 1
```

Level 2-E: Corona

System Statistics

66 Nodes, 2x AMD Opteron 8-core 6128 (16 cores/node), 32 or 64GB RAM per node. 1 or 2TB scratch space per node. Hostname is corona.cc.lehigh.edu.

Scheduling

Job scheduling using PBS Torque/Maui is required for all jobs.

Queues

Jobs may be submitted to bf, s-short, s, p, or p-ib queues depending on requirements.

Corona Queues

Queue Name	Priority	Job Node Limit			Job Wall-Clock Limit	User Node Limit			Queue Node Limit			Comments
		IB	non-IB	Total		IB	non-IB	Total	IB	non-IB	Total	
s-short	800,000	1	1	1	1 hour	2	2	2	2	2	2	non-IB nodes allocated before IB
p-ib	600,000	24	0	24	168 node-days	24	0	24	24	0	24	non-IB nodes allocated before IB
p	400,000	12	36	48	336 node-days	12	36	48	12	36	48	non-IB nodes allocated before IB
s	200,000	1	1	1	120 hours	16	16	32	12	38	50	non-IB nodes allocated before IB
bf	1	1	1	1	6 hours	22	40	62	22	40	62	non-IB nodes allocated before IB

Using Corona

- Compile and submit jobs from the head node; don't run jobs on corona1, they will be killed.
- Submit jobs early and often. If the resource is busy, Maui can't hold a spot if you haven't submitted any jobs.
- Don’t grossly overestimate walltimes, especially if you submit the same type of job regularly.
- Remember to cleanup /scratch if you use it before your job completes. This space is erased at the beginning of each new job.

http://corona.cc.lehigh.edu/c/CoronaAccessCookbook.pdf

PBS Sample

```
# This is a comment
#PBS -N xhpl
#PBS -q p-ib
#PBS -l nodes=10:ppn=16,walltime=72:00:00
#PBS -M sma310@lehigh.edu
#PBS -m bea
#PBS -e HPL.err
#PBS -o HPL.out
echo "Start MPICH PBS job          : master node `hostname`, date `date`"
echo "PBS job id      : $PBS_JOBID"
echo "PBS_O_WORKDIR           : $PBS_O_WORKDIR"
NPROCS=`wc -l < $PBS_NODEFILE`
echo "Number of requested nodes: $NPROCS"
echo "Assigned node names: `cat $PBS_NODEFILE`"
executable=xhpl
echo "executable name       : $executable"
/home/local/bin/mpirun-hydra -machinefile $PBS_NODEFILE -np $NPROCS $PBS_O_WORKDIR/$executable
echo "End MPICH PBS job           : master node `hostname`, date `date`"
```

mpirun vs. mpirun-hydra

- The mpirun command uses the MPD ring to allow nodes to communicate with each other.
- Works, but can be fragile. All users share one ring; if it crashes, it breaks node communication for everyone!
- Deprecated by Argonne National Lab; should avoid using if possible.

```
mpirun -machinefile $PBS_NODEFILE -np $NPROCS $executable
```

- The mpirun-hydra command uses the Hydra process manager to facilitate node communication.
- Resource manager provided by ANL to replace MPD; more robust.
- Allows inter-node communication without relying on the MPD ring.

```
/home/local/bin/mpirun-hydra -machinefile $PBS_NODEFILE -np $NPROCS
    $PBS_O_WORKDIR/$executable
```

Queue and Job Status

- qsub file.submit: submit the job described in file.submit.
- qstat: show jobs you have submitted to the queue.
- qdel jobID: removes jobID from the queue.
- checkjob/tracejob jobID: check the status of a submitted/running job or show historical information (past 24 hours).
- showstart jobID: show computed start time for job.
- myQuota: display your current quota usage and information in an understandable format.
- http://corona.cc.lehigh.edu/q : snapshots of the queue, updated every 15 minutes.

Fairshare

- Measure of system utilization (proc-hours) which is used to adjust job priority.
- Used to prevent a single user from monopolizing the system.
- 14 day window with a decay policy. Total FS usage determined from usage over that time, compared to FS target(per-user) and used to adjust priority.

- Note: If there are no other jobs in the queue, FS won't help, it does not prevent low priority jobs from running. This also means a low priority isn't always bad, only hurts when there is resource contention.

How to Get Help

Best way is to submit a ticket at:

http://www.lehigh.edu/help

Information to include:

- What you are trying to do, including software or compiler names and versions.
- What system you're using as a platform.
- What you've tried to do, and the result.
- Any logs you generated while working.