Bioinformatics Facility at the Biotechnology/Bioservices Center

Co-Heads : J.P. Gogarten, Paul Lewis

Facility Scientist : Pascal Lapierre

Hardware/Software Manager: Jeff Lary

Mandate of the Facility:

To provide computational power and technical support to both academia and industry. These services are available, free of charge, to faculty and students within the University system and are available at negotiated rates to other academic institutions

The usage of the cluster is for research purpose only.

The Hardware:

- 17 node Dell Linux cluster running Redhat EL5. Each compute node is equipped with 2 x Quad-core 2.53 GHz Intel Xeon processors and 32 GB of memory
- 18-node, 36 processor Apple Xserve Cluster
- Small Linux-based satellite cluster

Getting Started:

Help can be found on the Uconn Bioinformatics Wiki page :
http://137.99.47.91/wiki/index.php/Main_Page

To log on the servers :

PC:	MAC:
• Putty (SSH client)	• Console or Jellyfissh (SSH)
• FileZilla (SFTP)	• Fugu (SFTP)

bbcxsrv1.biotech.uconn.edu (Xserve cluster)
bbcsrv3.biotech.uconn.edu (Dell server)

Accounts:

- Username for accounts :
- Passwords are
- When you login for the first time, create a new password by typing : passwd

Unix basic commands:

- ls : List file of a directory
- cd : Change directory
- mkdir : make directory
- cp : copy
- mv : move
- more/less/cat : view content of a file
- man <commands> : manual of a command
- pwd : display current path
- up arrow : cycles through previous commands
- tab : autocompletions of file names

Unix advance commands:

- lsload (Dell only) : See the current CPU loads
- lslogin (Dell) or rlogin (Xserver) : log to a sub-node
- qstat : Display the status of the queue
- qsub : Submit a script to the queue (qsub perl run.pl)
- bjobs: Display the status of your jobs (if any)
- ssh compute-1-x (Dell) or ssh nodex (Xserver) : To manually log on a sub-node
- qdel : To terminate a job running on the queue
- ps ux: Display processes status

Programs and packages available:

- Beast
- BLAST Suite
- CLUSTALW
- DarkHorse
- EMBOSS
- FastTree
- GARLI
- gsAssembler
- HMMer
- HyPhy
- Mauve
- Mothur
- MrBayes
- MUSCLE
- MUMmer
- PAML
- nhPhyML
- Paup*
- Phycas
- PHYLIP
- Phylobayes
- PhyML
- RaxML

Usage etiquette:

- There are no official limits on the number of jobs you can run on the cluster but…please refrain from using all the nodes a the same time.
- NEVER run anything on the head node of a cluster (default login node).
- Keep track of what you are running and where so if something goes wrong, you can go back and kill the desired job.
- Each nodes have their own hard drive (scratch drive). It is advised when possible to run and write your output files on this drive (/scratch), then copy the file back to your home directory when done.

Running MPI Jobs on the Dell Cluster

-Everything you need is on the Wiki page with example shell scripts.

-128 available CPUs, used at about 50% in the last few weeks.

- What is MPI?

Message Passing Interface (MPI) is a specification for an API that allows many computers to communicate with one another.

At this time there are a few mpi-enabled applications on the Dell cluster (bbcsrv3):

- clustalw-mpi (ClustalW-MPI v0.13, based on ClustalW v1.82)

- mb-mpi (MrBayes v3.1.2)

- phyml-mpi (Phyml v3.0)

- R (2.9.2)

- mpi-blast

Before you run your first MPI Job, make sure that the following code is in your .bashrc file:

```
# Load saved modules
module load mpi/openmpi-interconnects-gnu
```

Shell script for Clustalw-MPI

```
#!/bin/bash
 #BSUB -q normal                    # submit the job to the normal queue, which is the default queue
 #BSUB -o /home/<yourusername>/clustalw-mpi-%J.o   # name the output file; %J inserts the current job number
 #BSUB -e /home/<yourusername>/clustalw-mpi-%J.e   # name the error file; %J inserts the current job number
 #BSUB -J mpi-job                   # give the job a jobname, mpi-job
 #BSUB -n 4                         # define the number of processors to use
 #BSUB -a openmpi                   # define the type of MPI to use.
 #
 cd /home/<yourusername>/clustalw-mpi-0.13/
 #
 mpirun.lsf --mca btl ^openib ./clustalw-mpi -infile=CFTR.input  -newtree=CFTR.mytree
 #                                  # the --mca btl ^openib part of the command line is telling mpirun
 #                                  # to exclude using infiniband in the byte transfer layer (btl)
```

Bsub < shell_script.sh

Basic Bsub scripts :

```
#!/bin/bash
#BSUB -o /home/<USERNAME>/outputfile  # Put output and errors in file outputfile
#
cd /home/<USERNAME>/<working directory>/
phyml -i somefile.phy -d nt -b 20 -m JC69 -v 0 -c 4 -a e -s BEST -o tlr
```

Bsub scripts to run on scratch drive :

```
#! /bin/bash
 #BSUB -B  # Send mail at beginning of job execution
 #BSUB -N  # Send mail at end of job execution
 #BSUB -u Firstname.Lastname@uconn.edu # users email destination
 #BSUB -J myjob  # Give the job the name 'myjob'
 #BSUB -o outputfile  # Put output and errors in file outputfile

 if [ ! -d "/scratch/$USER" ]; then
 mkdir /scratch/$USER
 fi
 #
 if [ ! -d "/scratch/$USER/subdirname" ]; then
 mkdir /scratch/$USER/subdirname
 fi
 #
 cd /scratch/$USER/subdirname  # cd to the working directory in the scratch area
 cp $HOME/myprog .
 cp $HOME/data/inputfile1 .
 ./myprog < inputfile1 > outputfile1
 #
 cp outputfile1 $HOME/savedir  # make sure that $HOME/savedir exists!
 #
 rm -f /scratch/$USER/subdirname/*
```

Useful Script if you want to distribute Analysis over multiple CPUs

```
#BSUB -J test2[1-575]%40  #Will cycle from 1 to 575 and start processes over a maximum of 40 processors are reached

#BSUB -o bootjob%J.log  #Create log files named bootjob1.log to bootjob475.log

#BSUB $LSB_JOBINDEX

perl ~/map_algor/bootstraps/boot_sphere.pl $LSB_JOBINDEX
```