Genome Center Clusters

UCDAVIS genome center

Introduction

“If you were plowing a field, which would you rather use: Two strong oxen or 1024 chickens?”

–Seymour Cray

Introduction

- What is a cluster?

Introduction

- What is a cluster?
  - A collection of machines that work together

Introduction

- What is a cluster?
  - A collection of machines that work together
- Why use a cluster?

Introduction

- What is a cluster?
    - A collection of machines that work together
- Why use a cluster?
    - Chickens are cheaper than oxen

Introduction

- What is a cluster?
  - A collection of machines that work together
- Why use a cluster?
  - Chickens are cheaper than oxen
  - and easier to feed

Introduction

- What is a cluster?
  - A collection of machines that work together
- Why use a cluster?
  - Chickens are cheaper than oxen
  - and easier to feed
  - ...but try making 1024 chickens move in the same direction

Clusterable Jobs

- Parallel

Clusterable Jobs

- Parallel
  - fine-grained parallel

Clusterable Jobs

- Parallel
  - fine-grained parallel
  - course-grained parallel

Clusterable Jobs

- Parallel
  - fine-grained parallel
  - course-grained parallel
  - embarrassingly parallel

Clusterable Jobs

- Parallel
  - fine-grained parallel
  - course-grained parallel
  - embarrassingly parallel
- Serial

Clusterable Jobs

- Parallel
  - fine-grained parallel
  - course-grained parallel
  - embarrassingly parallel
- Serial
  - Can still benefit from a cluster environment

Available Resources

- genbeo

Available Resources

- genbeo
- shiraz

Available Resources

- genbeo
- shiraz
- apple

Genbeo

- 19 nodes

Genbeo

- 19 nodes
- Dual Opteron

Genbeo

- 19 nodes
- Dual Opteron
- 4G memory per node

Genbeo

- 19 nodes
- Dual Opteron
- 4G memory per node
- Gigabit ethernet interconnect

Genbeo

- 19 nodes
- Dual Opteron
- 4G memory per node
- Gigabit ethernet interconnect
- 3.2TB storage

Genbeo

- 19 nodes
- Dual Opteron
- 4G memory per node
- Gigabit ethernet interconnect
- 3.2TB storage
  - 800G User home dirs and mysql

Genbeo

- 19 nodes
- Dual Opteron
- 4G memory per node
- Gigabit ethernet interconnect
- 3.2TB storage
  - 800G User home dirs and mysql
  - 800G Shared applications

Genbeo

- 19 nodes
- Dual Opteron
- 4G memory per node
- Gigabit ethernet interconnect
- 3.2TB storage
  - 800G User home dirs and mysql
  - 800G Shared applications
  - 800G Shared filesystem

Genbeo

- 19 nodes
- Dual Opteron
- 4G memory per node
- Gigabit ethernet interconnect
- 3.2TB storage
  - 800G User home dirs and mysql
  - 800G Shared applications
  - 800G Shared filesystem
  - 800G backups

Shiraz

- 110 nodes

Shiraz

- 110 nodes
- Dual processor, dual core Opteron (4 cores/node)

Shiraz

- 110 nodes
- Dual processor, dual core Opteron (4 cores/node)
- 32 nodes with 8G, remainder with 4G

Shiraz

- 110 nodes
- Dual processor, dual core Opteron (4 cores/node)
- 32 nodes with 8G, remainder with 4G
- Gigabit ethernet interconnect

Shiraz

- 110 nodes
- Dual processor, dual core Opteron (4 cores/node)
- 32 nodes with 8G, remainder with 4G
- Gigabit ethernet interconnect
- 4.4TB attached storage

Shiraz

- 110 nodes
- Dual processor, dual core Opteron (4 cores/node)
- 32 nodes with 8G, remainder with 4G
- Gigabit ethernet interconnect
- 4.4TB attached storage
  - 1.2T User storage

Shiraz

- 110 nodes
- Dual processor, dual core Opteron (4 cores/node)
- 32 nodes with 8G, remainder with 4G
- Gigabit ethernet interconnect
- 4.4TB attached storage
    - 1.2T User storage
    - 1.6T shared databases (PDB, blastdb, ...)

Shiraz

- 110 nodes
- Dual processor, dual core Opteron (4 cores/node)
- 32 nodes with 8G, remainder with 4G
- Gigabit ethernet interconnect
- 4.4TB attached storage
  - 1.2T User storage
  - 1.6T shared databases (PDB, blastdb, ...)
  - 1.6T Shared filesystem

Apple

- 37 nodes

Apple

- 37 nodes
- Dual PowerPC G4

Apple

- 37 nodes
- Dual PowerPC G4
- 1-2G memory per node

Apple

- 37 nodes
- Dual PowerPC G4
- 1-2G memory per node
- Gigabit ethernet interconnect

Apple

- 37 nodes
- Dual PowerPC G4
- 1-2G memory per node
- Gigabit ethernet interconnect
- 1.6TB attached storage

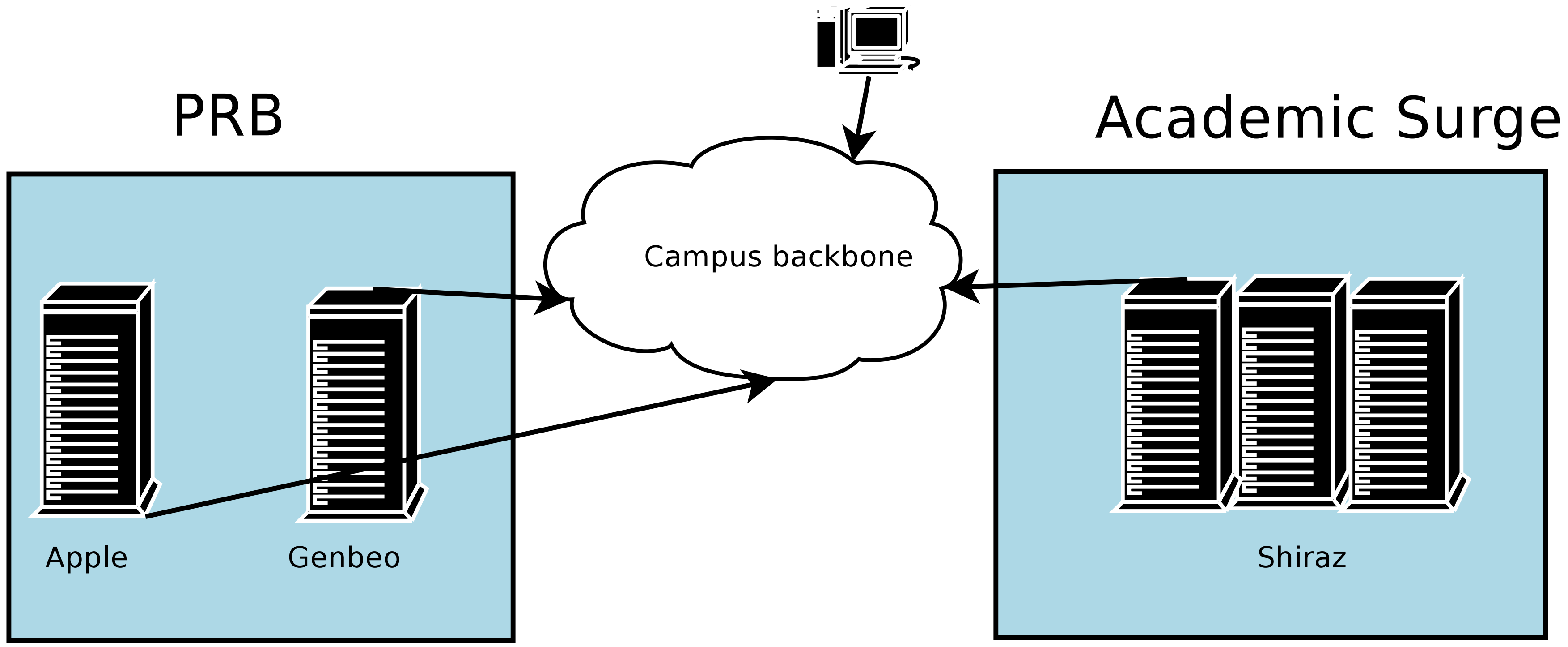
Apple

- 37 nodes
- Dual PowerPC G4
- 1-2G memory per node
- Gigabit ethernet interconnect
- 1.6TB attached storage
    - 800G User home directories

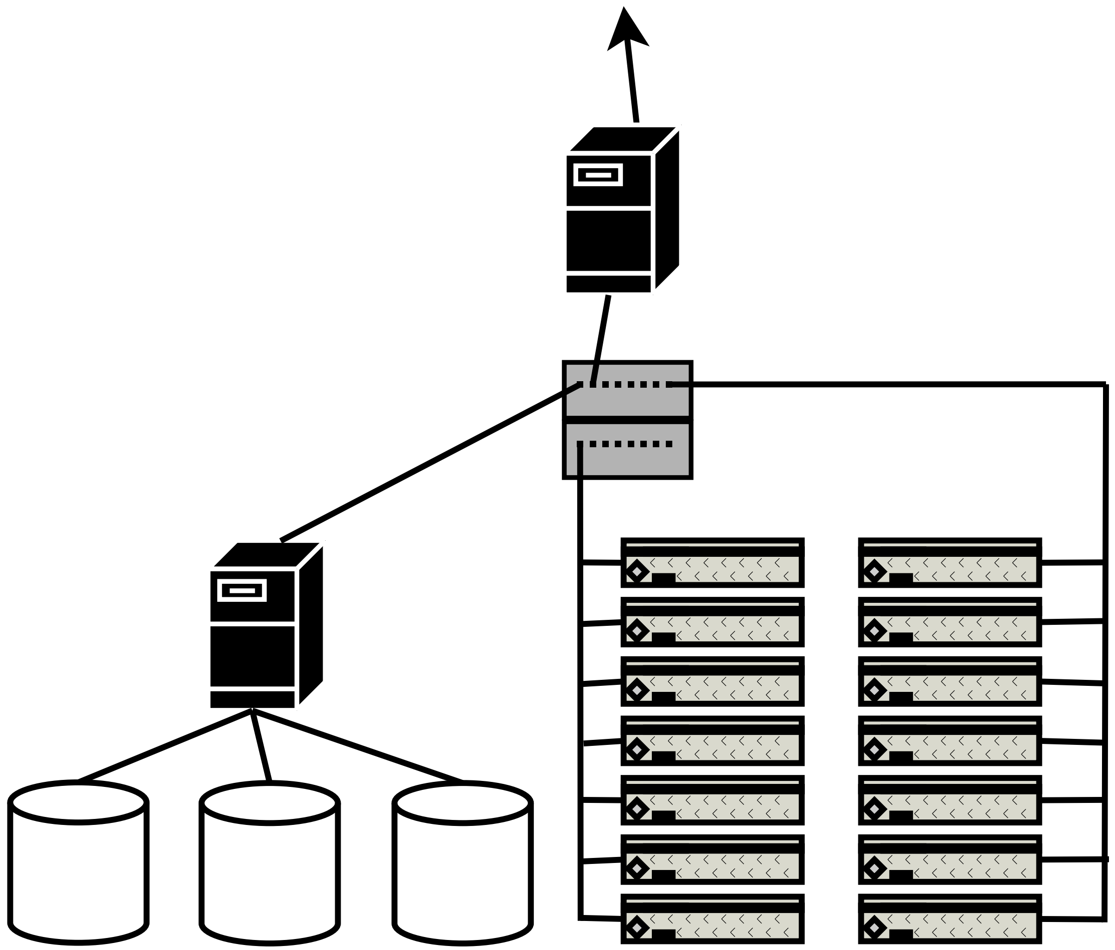
Apple

- 37 nodes
- Dual PowerPC G4
- 1-2G memory per node
- Gigabit ethernet interconnect
- 1.6TB attached storage
  - 800G User home directories
  - 800G Mysql and postgres databases

The Big Picture



Anatomy of a Cluster



Available Software

- Biological databases:

Available Software

- Biological databases:
  - pdb

Available Software

- Biological databases:
  - pdb
  - blastdb

Available Software

- Biological databases:
  - pdb
  - blastdb
  - ..

Available Software

Compilers

- Gnu compiler suite

Available Software

Compilers

- Gnu compiler suite
- Pathscale floating license

Available Software

Compilers

- Gnu compiler suite
- Pathscale floating license
- Intel?

Available Software

- biopython
- clustalw
- EMBOSS
- BLAST
- HMMER
- mrbayes
- t-coffee
- phylip
- ...

Available Software

- R

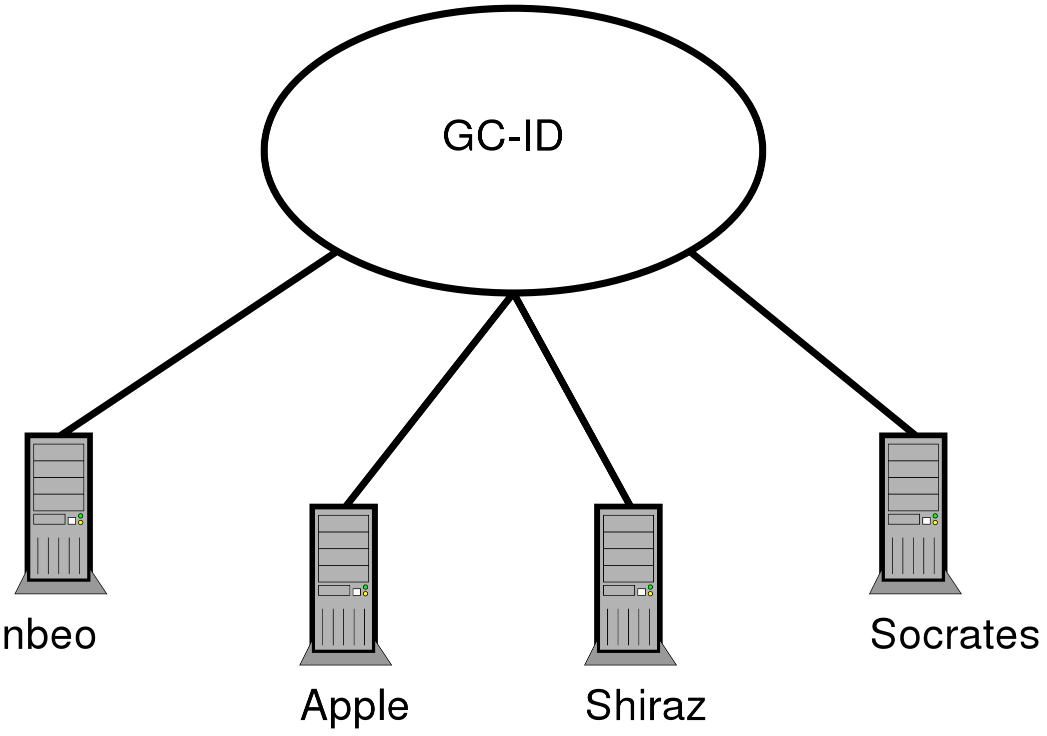
Available Software

- R
- mysql

Available Software

- R
- mysql
- Request your app!

Your Cluster Account



Same username/password works on all hosts on which you have an account.

Accessing the cluster

- Access using ssh

Accessing the cluster

- Access using ssh
- Transfer files using scp/sftp/rsync

Accessing the cluster

- Access using ssh
- Transfer files using scp/sftp/rsync
- Web interface

Accessing the cluster: ssh

- `ssh username@genbeo.genomecenter.ucdavis.edu`

Accessing the cluster: ssh

- `ssh username@genbeo.genomecenter.ucdavis.edu`
- Add `-X` option (capital!) to forward X connections

Accessing the cluster: ssh

- `ssh username@genbeo.genomecenter.ucdavis.edu`
- Add `-X` option (capital!) to forward X connections
- Clients for windows:

Accessing the cluster: ssh

- `ssh username@genbeo.genomecenter.ucdavis.edu`
- Add `-X` option (capital!) to forward X connections
- Clients for windows:
  - putty (free!)

Accessing the cluster: ssh

- `ssh username@genbeo.genomecenter.ucdavis.edu`
- Add `-X` option (capital!) to forward X connections
- Clients for windows:
  - putty (free!)
  - SecureCRT

Accessing the cluster: ssh

- `ssh username@genbeo.genomecenter.ucdavis.edu`
- Add `-X` option (capital!) to forward X connections
- Clients for windows:
  - putty (free!)
  - SecureCRT
  - ssh.com

Accessing the cluster: ssh

- `ssh username@genbeo.genomecenter.ucdavis.edu`
- Add `-X` option (capital!) to forward X connections
- Clients for windows:
  - putty (free!)
  - SecureCRT
  - ssh.com
  - winscp

Accessing the cluster: copying data

- scp

Accessing the cluster: copying data

- scp
  - `scp file username@genbeo:`

Accessing the cluster: copying data

- scp
  - `scp file username@genbeo:`
- rsync

Accessing the cluster: copying data

- scp
  - `scp file username@genbeo:`
- rsync
  - `rsync -a directory username@genbeo:`

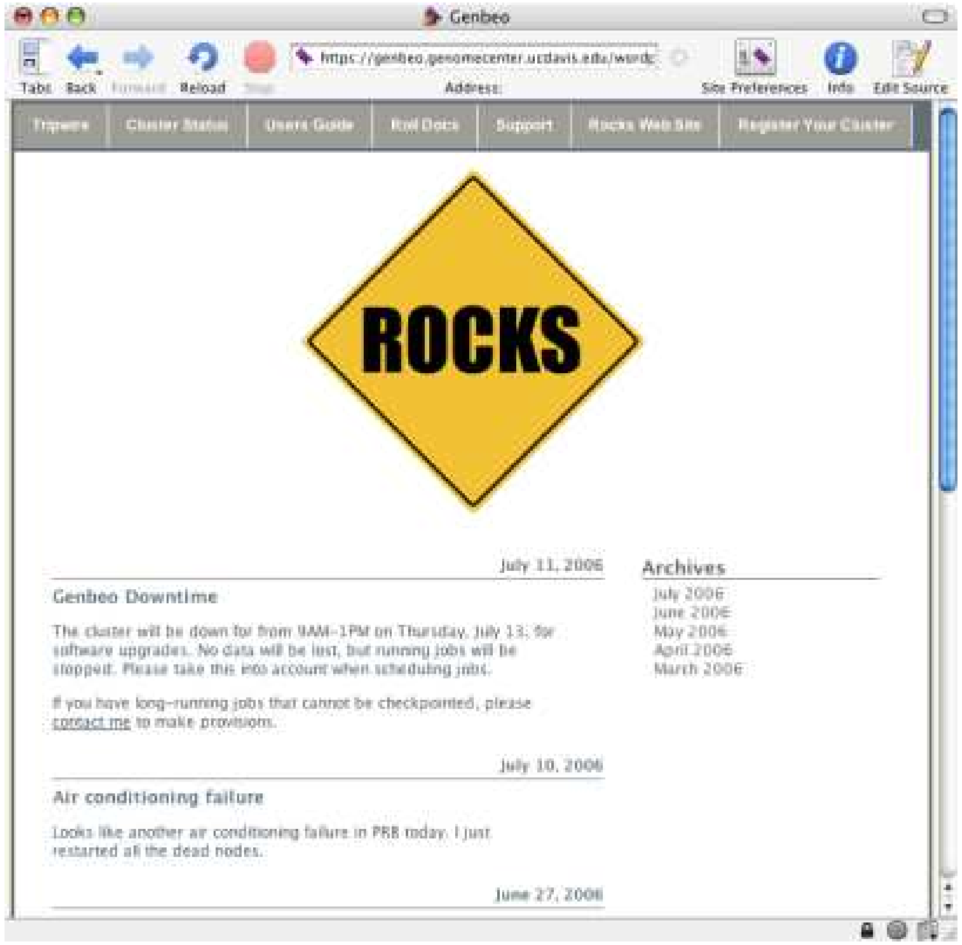
Accessing the cluster: copying data

- scp
  - `scp file username@genbeo:`
- rsync
  - `rsync -a directory username@genbeo:`
- sftp

Accessing the cluster: copying data

- scp
  - `scp file username@genbeo:`
- rsync
  - `rsync -a directory username@genbeo:`
- sftp
  - ftp-like interface that works over ssh

Cluster Web Interface



July 11, 2006

Genbeo Downtime

The cluster will be down for from 9AM–1PM on Thursday, July 13, for software upgrades. No data will be lost, but running jobs will be stopped. Please take this into account when scheduling jobs.

If you have long-running jobs that cannot be checkpointed, please contact me to make provisions.

July 10, 2006

Air conditioning failure

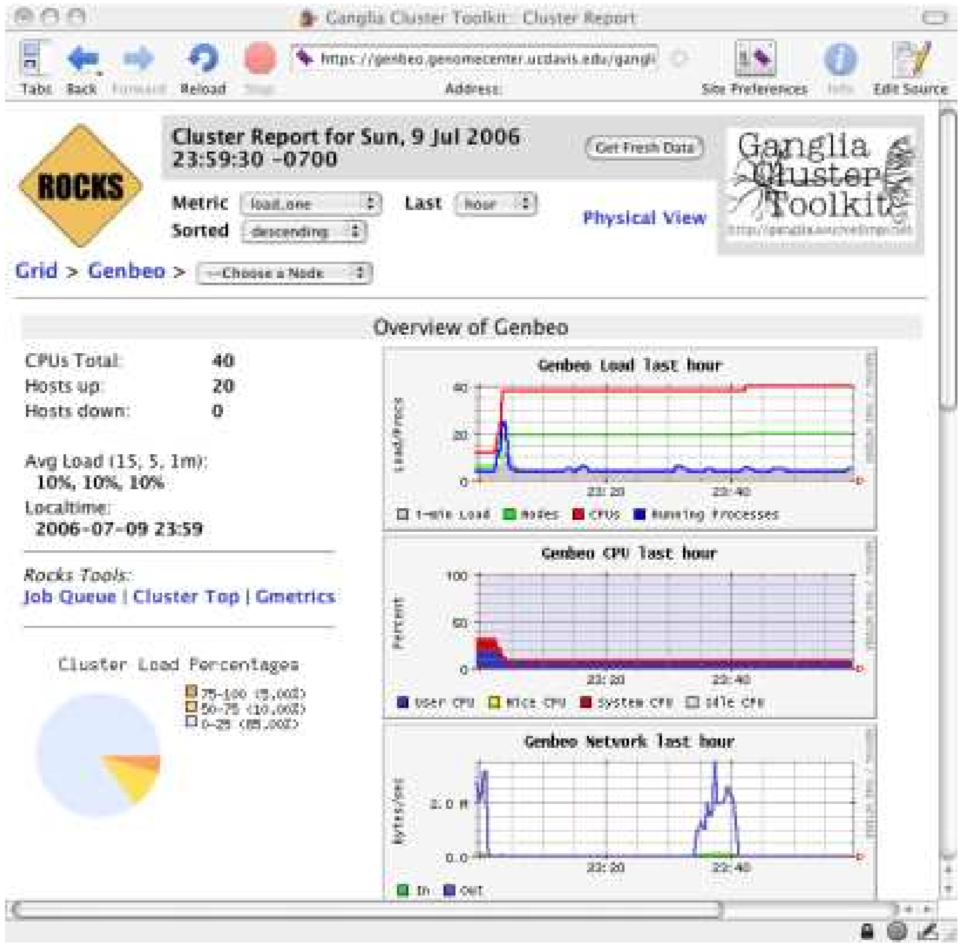
Looks like another air conditioning failure in PRB today. I just restarted all the dead nodes.

June 27, 2006

Archives

July 2006
June 2006
May 2006
April 2006
March 2006

Cluster Web Interface



Ganglia Cluster Toolkit: Cluster Report

https://genbeo.genomecenter.ucdavis.edu/gangli

Cluster Report for Sun, 9 Jul 2006 23:59:30 -0700

Metric load_one Last hour
Sorted descending

Physical View

Grid > Genbeo > --Choose a Node

Overview of Genbeo

CPUs Total:	40
Hosts up:	20
Hosts down:	0

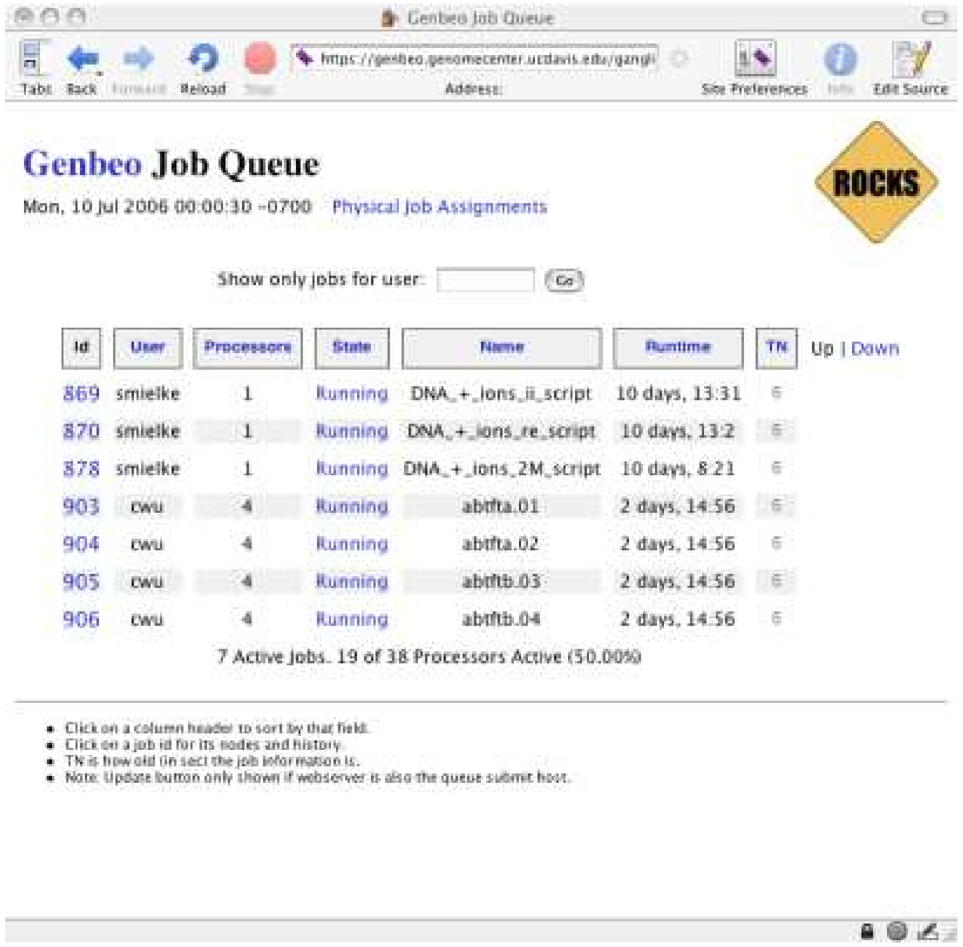
Avg Load (15, 5, 1m):
10%, 10%, 10%

Localtime:
2006-07-09 23:59

Rocks Tools:
Job Queue | Cluster Top | Gmetrics

Cluster Load Percentages

Cluster Web Interface



Genbeo Job Queue

Mon, 10 Jul 2006 00:00:30 -0700 Physical Job Assignments

Show only jobs for user: [ ] Go

Id	User	Processors	State	Name	Runtime	TN
869	smielke	1	Running	DNA_+_ions_ii_script	10 days, 13:31	6
870	smielke	1	Running	DNA_+_ions_re_script	10 days, 13:2	6
878	smielke	1	Running	DNA_+_ions_2M_script	10 days, 8:21	6
903	cwu	4	Running	abtfta.01	2 days, 14:56	6
904	cwu	4	Running	abtfta.02	2 days, 14:56	6
905	cwu	4	Running	abtftb.03	2 days, 14:56	6
906	cwu	4	Running	abtftb.04	2 days, 14:56	6

Up | Down

7 Active Jobs. 19 of 38 Processors Active (50.00%)

- Click on a column header to sort by that field.
- Click on a job id for its nodes and history.
- TN is how old (in secs) the job information is.
- Note: Update button only shown if webserver is also the queue submit host.

Sun Grid Engine

- What is a batch queue?

Sun Grid Engine

- What is a batch queue?
  - Manage cluster resources

Sun Grid Engine

- What is a batch queue?
  - Manage cluster resources
- Why use the batch queue?

Sun Grid Engine

- What is a batch queue?
  - Manage cluster resources
- Why use the batch queue?
  - Share cluster resources

Sun Grid Engine

- What is a batch queue?
  - Manage cluster resources
- Why use the batch queue?
  - Share cluster resources
  - You don’t need to worry about when/where your jobs run

Sun Grid Engine

- What is a batch queue?
  - Manage cluster resources
- Why use the batch queue?
  - Share cluster resources
  - You don’t need to worry about when/where your jobs run
  - Submit a whole bunch of jobs and go home!

SGE Terminology: slots

- A resource allocated to your job

SGE Terminology: slots

- A resource allocated to your job
- We define one slot per CPU core

SGE Terminology: slots

- A resource allocated to your job
- We define one slot per CPU core
- Can request number of slots when you submit job

SGE Terminology: Parallel Environm

- Need to use a PE when you want more than one slot

SGE Terminology: Parallel Environm

- Need to use a PE when you want more than one slot
- PE can specify start and stop programs (eg mpirun)

SGE Terminology: Parallel Environm

- Need to use a PE when you want more than one slot
- PE can specify start and stop programs (eg mpirun)
- PEs avilable:

SGE Terminology: Parallel Environm

- Need to use a PE when you want more than one slot
- PE can specify start and stop programs (eg mpirun)
- PEs avilable:
  - serial

SGE Terminology: Parallel Environm

- Need to use a PE when you want more than one slot
- PE can specify start and stop programs (eg mpirun)
- PEs avilable:
  - serial
  - mpich

SGE Terminology: Job Array

- Run the same job multiple times

SGE Terminology: Job Array

- Run the same job multiple times
- submit/manage as a single job

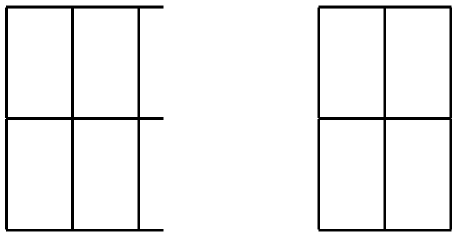
SGE Terminology: Job Array

- Run the same job multiple times
- submit/manage as a single job
- Ideal for running the same program repeatedly with different input files or parameters

Queue example

Job	Slots
A	6
B_1	1
B_2	1
B_3	1
C	4

Queue



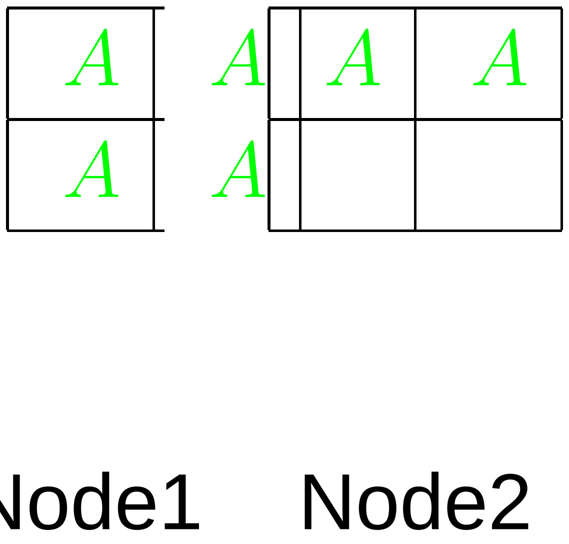
Node1 Node2

Three jobs waiting in queue...

Queue example

Job	Slots
A	6
B_1	1
B_2	1
B_3	1
C	4

Queue

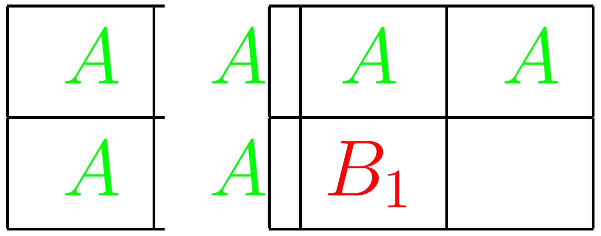


Job A is scheduled

Queue example

Job	Slots
A	6
B_1	1
B_2	1
B_3	1
C	4

Queue



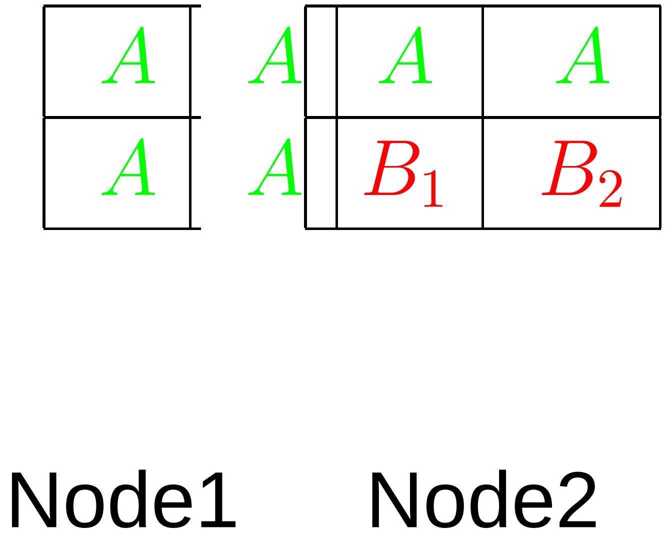
Node1 Node2

Job B_1 is scheduled

Queue example

Job	Slots
A	6
B_1	1
B_2	1
B_3	1
C	4

Queue

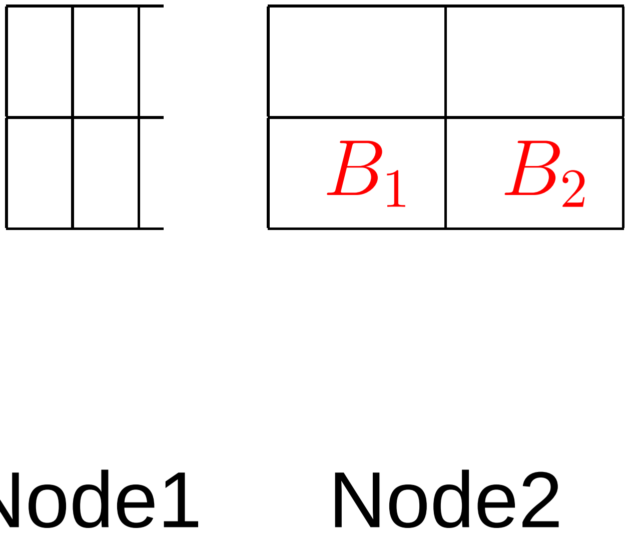


Job B_2 is scheduled

Queue example

Job	Slots
B_1	1
B_2	1
B_3	1
C	4

Queue

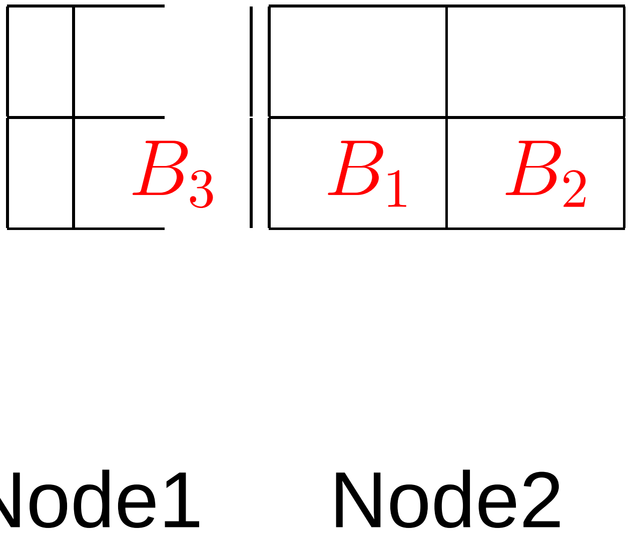


Job A finishes

Queue example

Job	Slots
B_1	1
B_2	1
B_3	1
C	4

Queue

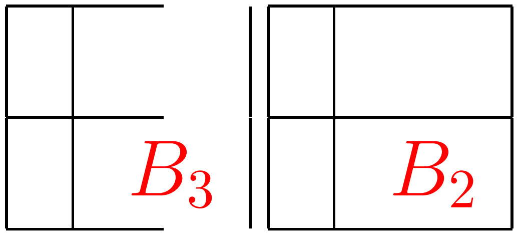


Job B_3 is scheduled

Queue example

Job	Slots
B_2	1
B_3	1
C	4

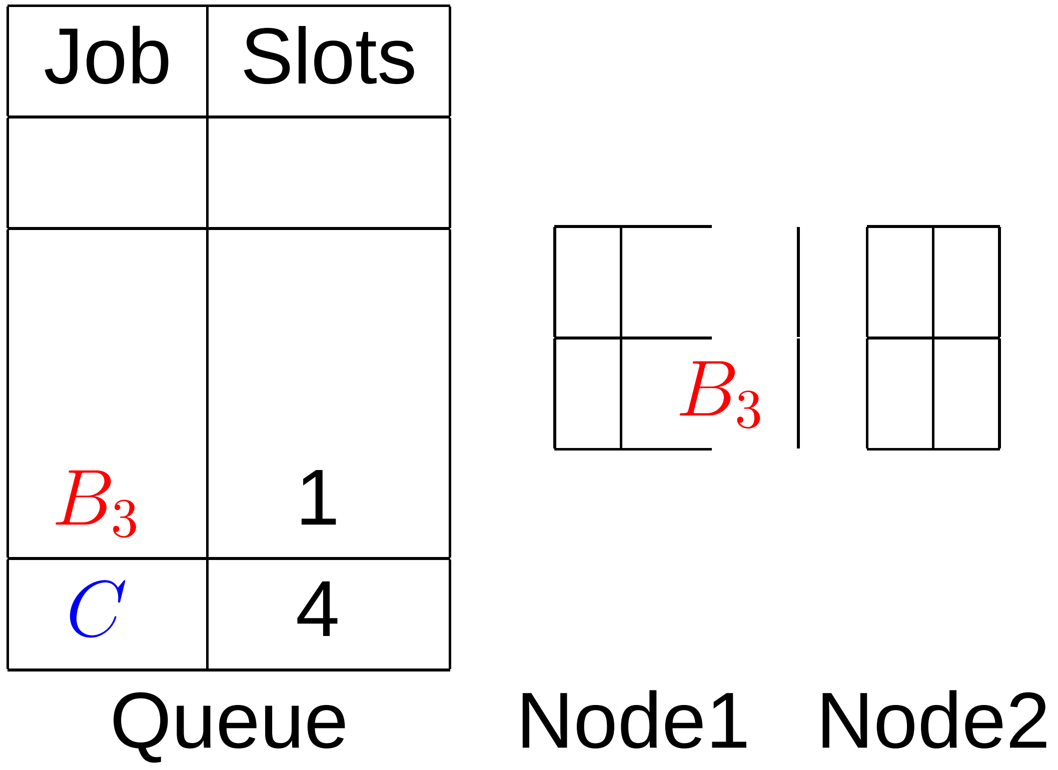
Queue



Node1 Node2

Job B_1 finishes

Queue example

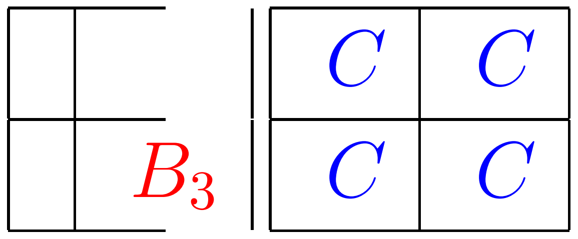


Job B_2 finishes

Queue example

Job	Slots
B_3	1
C	4

Queue



Node1 Node2

Job C is scheduled

SGE Commands

- qsub: submit jobs

SGE Commands

- qsub: submit jobs
- qstat: get job status

SGE Commands

- qsub: submit jobs
- qstat: get job status
- qdel: remove a job

SGE Commands

- qsub: submit jobs
- qstat: get job status
- qdel: remove a job
- qlogin: interactive login

SGE Commands: qsub

Use the qsub command to submit a batch job to the system

Simplest case:

```
$ qsub file.sh
Your job 929 ("file.sh") has been submitted.
```

SGE Commands: qstat

Use the -f flag to see all jobs running...

```
$ qstat -f
queuename                     qtype used/tot. load_avg arch          states
----------------------------------------------------------------------------
all.q@compute-0-1.local       BIP   4/4       1.83     lx26-amd64
  2455 0.60500 proAwt     cwu          r     06/21/2006 12:06:06    4
----------------------------------------------------------------------------
all.q@compute-0-10.local      BIP   0/4       0.00     lx26-amd64    d
----------------------------------------------------------------------------
all.q@compute-0-98.local      BIP   4/4       2.80     lx26-amd64
  2823 0.51386 ccr5_SCH_A twang        r     07/07/2006 15:12:51    2
  2865 0.50500 rungb5b    xjdeng       r     07/08/2006 17:37:06    1
  2944 0.52905 g2l_ff03   zxwang       r     07/09/2006 22:07:21    1
----------------------------------------------------------------------------
all.q@compute-0-99.local      BIP   0/4       0.00     lx26-amd64
```

SGE Commands: qstat

...as well as those waiting to be run.

```
############################################################################
PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS - PENDING JOBS
############################################################################
2947 0.52905 g2f_ff03   zxwang       qw    07/09/2006 22:11:04    20
2948 0.52905 g2h_ff03   zxwang       qw    07/09/2006 22:11:55    20
2949 0.52905 g2q_ff03   zxwang       qw    07/09/2006 22:12:42    20
```

SGE Commands: qstat

Use the `-j <jobid>` flag to get more information about a job:

```
$ qstat -j 2947
job_number:                 2947
exec_file:                  job_scripts/2947
submission_time:            Sun Jul  9 22:11:04 2006
...
```

SGE Commands: qdel

- Use the `qdel` command to delete a previously scheduled job from the queue.

SGE Commands: qdel

- Use the `qdel` command to delete a previously scheduled job from the queue.
- Note: if the job was running, you may still have to kill the processes by hand.

SGE Commands: qdel

- Use the `qdel` command to delete a previously scheduled job from the queue.
- Note: if the job was running, you may still have to kill the processes by hand.
- The `-f` (force) option can sometimes be necessary to clean up jobs left behind, for example, if a node dies during the job.

SGE Commands: qlogin

Use the `qlogin` command to schedule an interactive login.

- Default is one slot

SGE Commands: qlogin

Use the `qlogin` command to schedule an interactive login.

- Default is one slot
- To allocate more slots, use the parallel environment serial and the number of slots
  ```
  qlogin -pe serial 2
  ```

SGE Commands: qlogin

Use the `qlogin` command to schedule an interactive login.

- Default is one slot
- To allocate more slots, use the parallel environment serial and the number of slots
  ```
  qlogin -pe serial 2
  ```
- Two slots on genbeo will give you the whole node (4 on shiraz)

SGE Commands: qlogin

Use the `qlogin` command to schedule an interactive login.

- Default is one slot
- To allocate more slots, use the parallel environment serial and the number of slots
  `qlogin -pe serial 2`
- Two slots on genbeo will give you the whole node (4 on shiraz)
- If enough slots are not avilable, `qlogin` will fail.

SGE Example: BLAST

Single Threaded single app (simple) BLAST (Basic Local Alignment Search Tool)

SGE script: (`sge_blast.sh`)

```
# Blast executable binary
BLAST=/opt/Bio/blast-2.2.14/bin/blastall

#$ -cwd
#$ -o blastout
#$ -e blasterr
#$ -q all.q
#$ -S /bin/bash

$BLAST $*
```

Lines with #$ are command-line flags for SGE.

SGE Example: BLAST

```
qsub -N serialBlast sge_blast.sh -p blastp -i query.aa
-d drosoph.aa -e 1e-40 -o output.blast
```

- `-N` Specify job name (`serialBlast`)

SGE Example: BLAST

```
qsub -N serialBlast sge_blast.sh -p blastp -i query.aa
-d drosoph.aa -e 1e-40 -o output.blast
```

- `-N` Specify job name (`serialBlast`)
- `-q` Specify queue in which to run job (`all.q`)

SGE Example: BLAST

```
qsub -N serialBlast sge_blast.sh -p blastp -i query.aa
-d drosoph.aa -e 1e-40 -o output.blast
```

- `-N` Specify job name (`serialBlast`)
- `-q` Specify queue in which to run job (`all.q`)
- `-S` Specify interpreter to run script (`/bin/bash`)

SGE Example: BLAST

```
qsub -N serialBlast sge_blast.sh -p blastp -i query.aa
-d drosoph.aa -e 1e-40 -o output.blast
```

- `-N` Specify job name (`serialBlast`)
- `-q` Specify queue in which to run job (`all.q`)
- `-S` Specify interpreter to run script (`/bin/bash`)
- `-o` Specify file for stdout of job

SGE Example: BLAST

```
qsub -N serialBlast sge_blast.sh -p blastp -i query.aa
-d drosoph.aa -e 1e-40 -o output.blast
```

- `-N` Specify job name (`serialBlast`)
- `-q` Specify queue in which to run job (`all.q`)
- `-S` Specify interpreter to run script (`/bin/bash`)
- `-o` Specify file for stdout of job
- `-cwd` Execute job in the directory from which it was submitted

SGE Example: mpi-BLAST

Parallel version of BLAST using MPI
SGE script: (`sge_mpiblast,sh`)

```
# Basic SGE script to run mpiblast

MPIRUN=/opt/mpich/gnu/bin/mpirun
MPIBLAST=/opt/Bio/mpiblast-1.4.0/bin/mpiblast

#$ -cwd
#$ -o mpiblastout
#$ -e mpiblasterr
#$ -q all.q
#$ -S /bin/bash

$MPIRUN -np $NSLOTS -machinefile /$TMPDIR/machines $MPIBLAST $*
```

SGE Example: mpi-BLAST

```
qsub -N mpiblast1 -pe mpich 4 sge_mpiblast.sh -p blastp
-d swissprot -i query.aa -o mpiblast.out -e 1e-40
```

- -pe Specify parallel environment and number of slots

SGE Example: R

- `-t 1-5:1`

SGE Example: R

- `-t 1-5:1`
- `-hold_jid job-name-or-id`

SGE Example: qlogin

Things to do

- Use the scheduler!

Things to do

- Use the scheduler!
- Checkpoint your job

Things to do

- Use the scheduler!
- Checkpoint your job
- Make use of local storage on the nodes for intermediate results

Things NOT to do

- Run jobs on the head node

Things NOT to do

- Run jobs on the head node
- Many simultaneous writes to network filesystem

Things NOT to do

- Run jobs on the head node
- Many simultaneous writes to network filesystem
- Go around scheduler and run directly on the nodes

The end

- Get an account: `http://genomecenter.ucdavis.edu/`, click on "Administration"
- Documentation and example scripts:

  `http://wiki.genomecenter.ucdavis.edu/bioinformatics/`
- Questions?