

INTRODUCTION TO HIGH PERFORMANCE COMPUTING (HPC)

*introduction to the gSTAR computing
cluster at Swinburne University*

Gregory B. Poole

Astronomy Data and Computing Services, Swinburne



*Based on content developed by
Amr H. Hassan*



TABLE OF CONTENTS

- Introduction to gSTAR - Hardware configuration
 - *Node layout*
 - *Storage*
- Logging in with SSH
- Environment modules
- Quotas
- The Portable Batch System (PBS)
- The Ganglia monitoring System

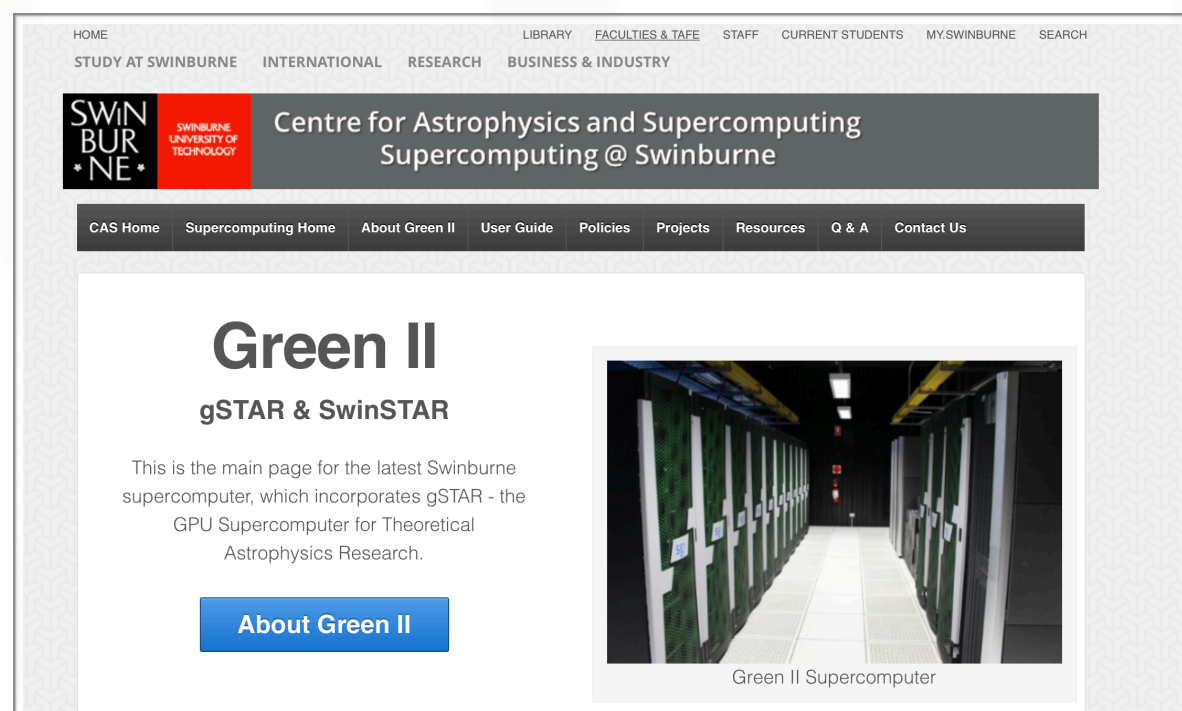
INTRODUCTION TO gSTAR – MORE HELP

If you need any additional help or more information:



... email the Swinburne HPC Help Desk:

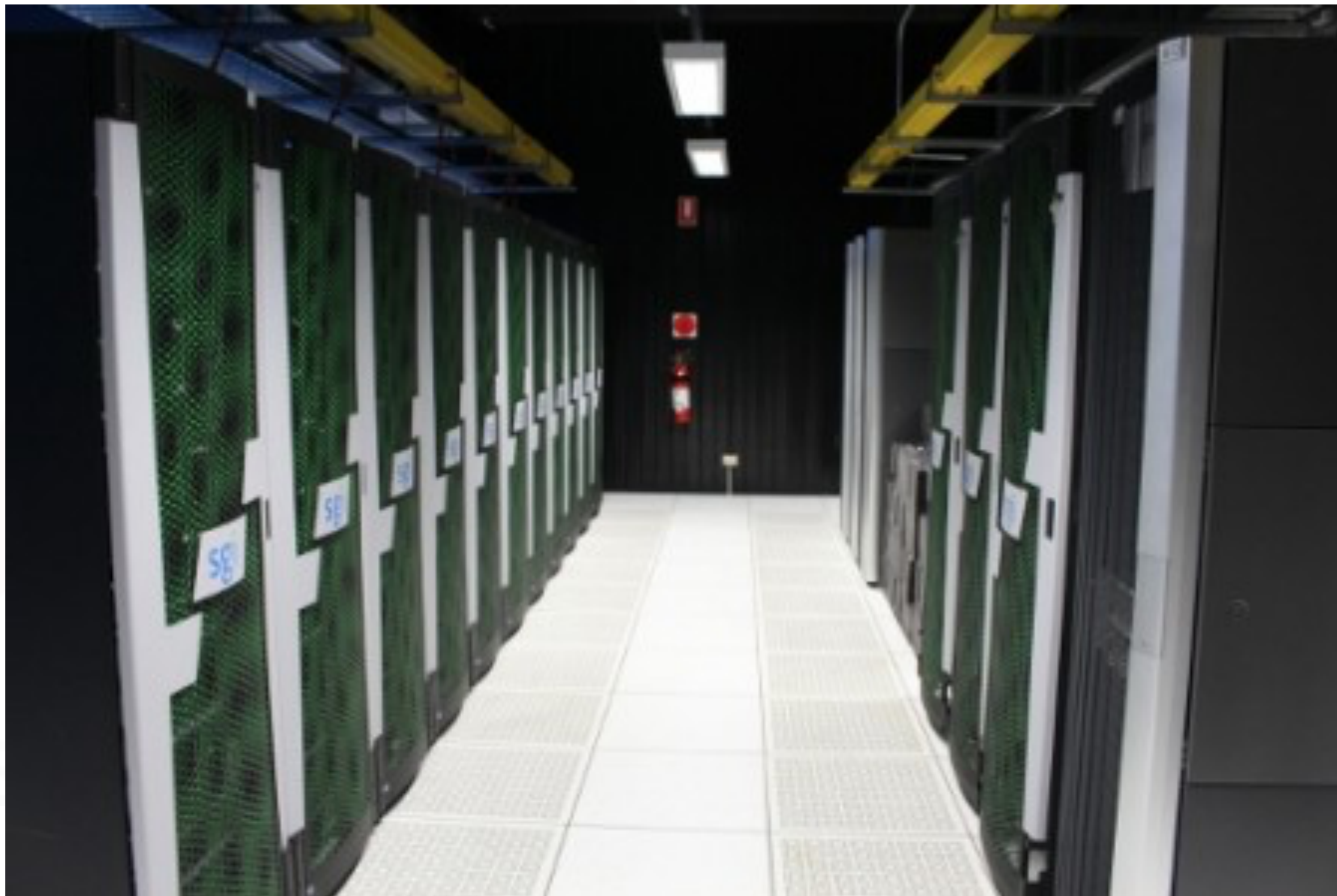
HPC-Support@swin.edu.au



... or visit the gSTAR webpage:

supercomputing.swin.edu.au

INTRODUCTION TO gSTAR – HARDWARE CONFIGURATION

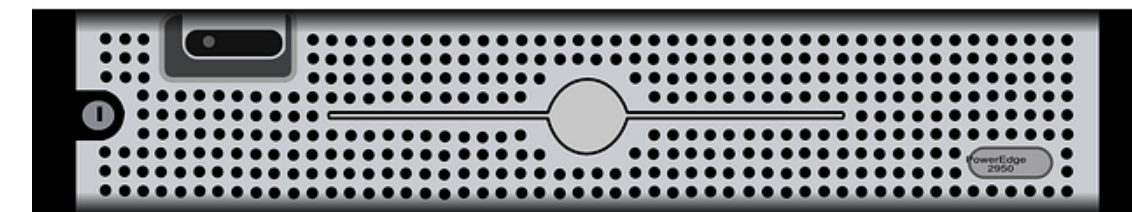


The gSTAR Cluster at Swinburne

INTRODUCTION TO gSTAR – HARDWARE CONFIGURATION – NODE LAYOUT

- Head Node (g2)

- Logging into this node gives access to all the others.



g2.hpc.swin.edu.au

- Interactive Nodes

- These 4 nodes are available for running small/quick test programs, etc.



gstar001

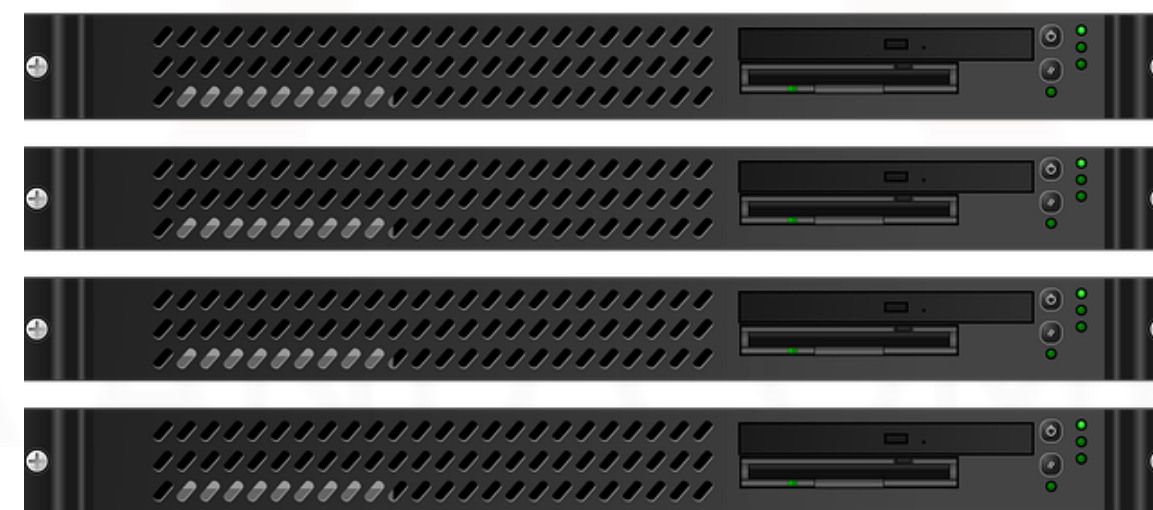
gstar002

sstar001

sstar002

- Compute Nodes

- These are the workhorse nodes of the system.
 - Accessed through a queue system.
 - Separated into two parts: gstar nodes and sstar nodes



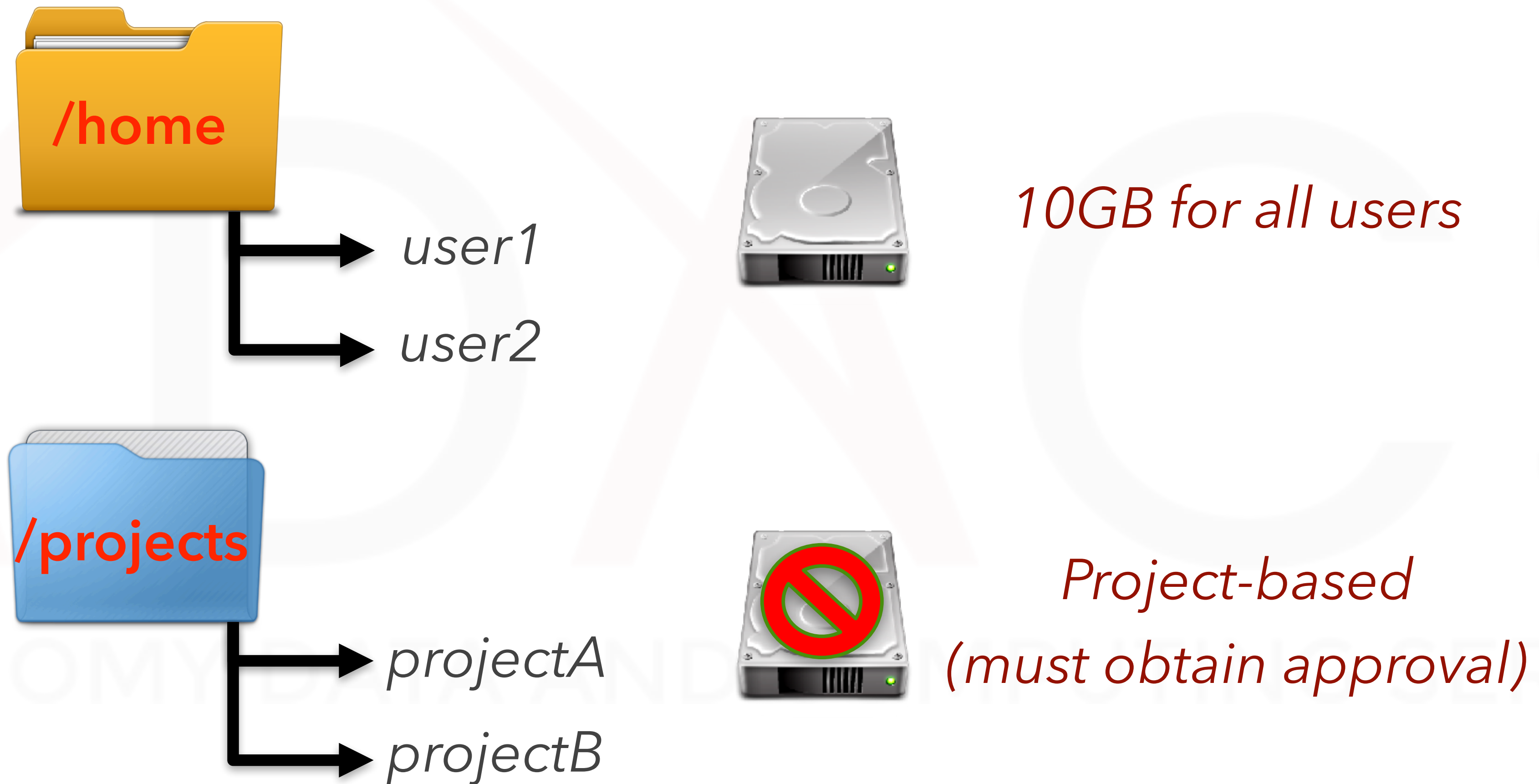
gstar###



sstar###

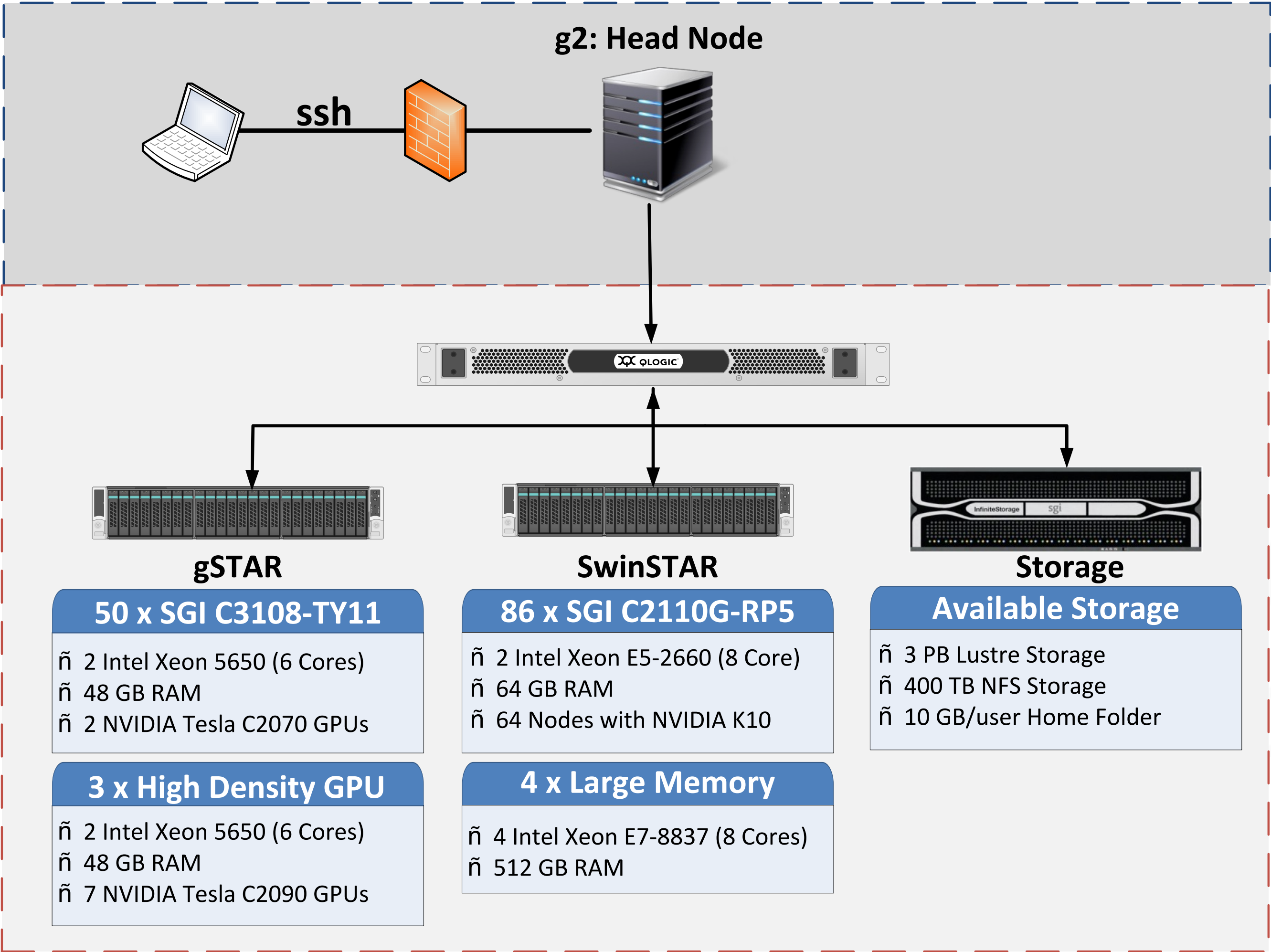
INTRODUCTION TO gSTAR – HARDWARE CONFIGURATION – STORAGE

Storage on gSTAR is split into a small user space and (optional) project spaces ...



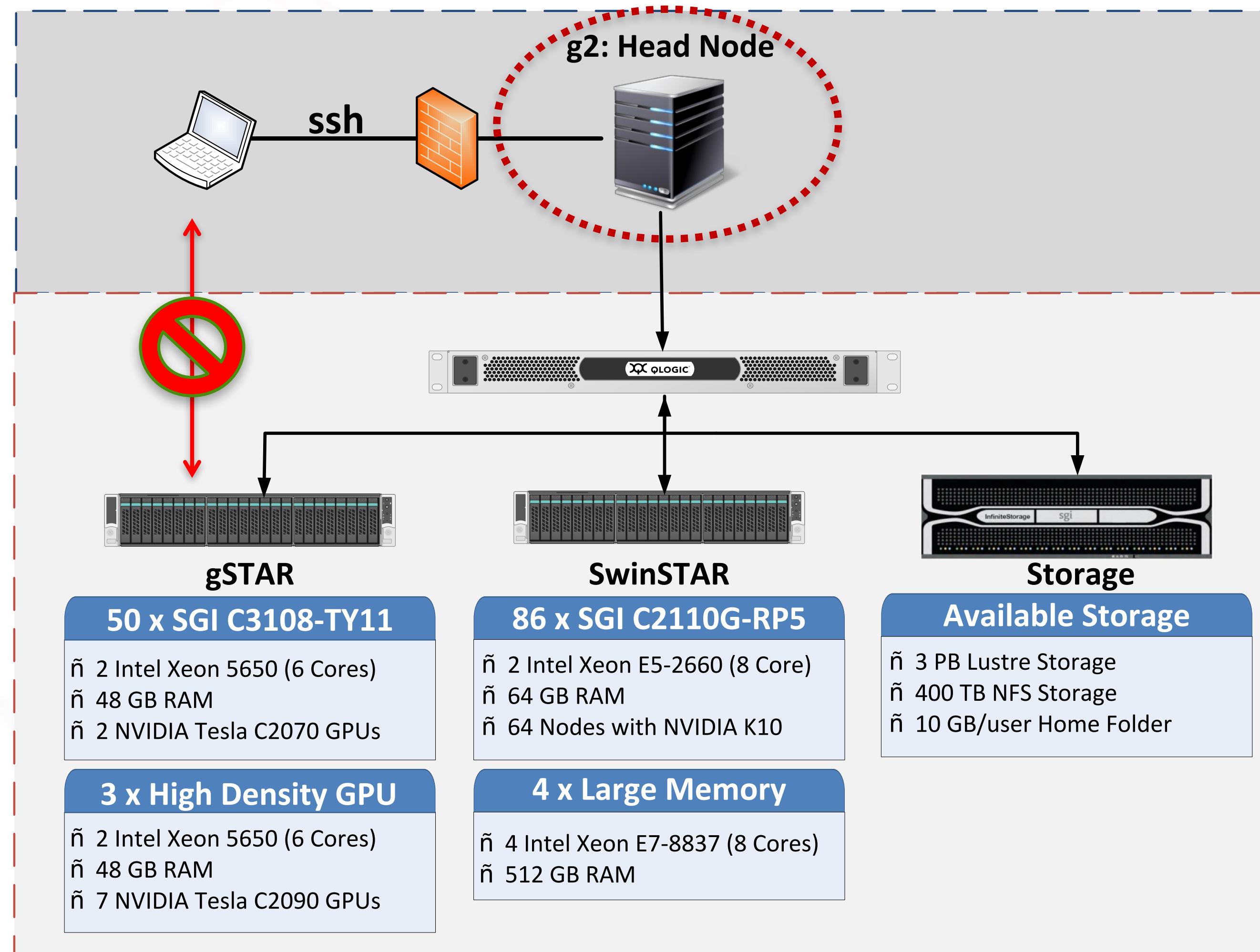
INTRODUCTION TO gSTAR – LOGGING IN w/ SSH

gSTAR must be accessed through the Secure Shell (SSH) protocol ...



INTRODUCTION TO gSTAR – LOGGING IN w/ SSH

Note that external access to the system must go through the "Head Node" ...



INTRODUCTION TO gSTAR – LOGGING IN w/ SSH

Every platform has several SSH tools to choose from. Some common defaults ...



... terminal (always available by default).



... Applications > Utilities > Terminal



... PuTTY: <http://goo.gl/XMggii>

INTRODUCTION TO gSTAR – LOGGING IN w/ SSH

Insert demonstration here

```
[terminal-3: ~] > ssh guest01@g2.hpc.swin.edu.au
guest01@g2.hpc.swin.edu.au's password:
Last login: Fri Jun 16 15:07:50 2017 from 136.186.24.21
-----
Wiki: http://supercomputing.swin.edu.au
-----
Please make sure you keep up-to-date with the usage policies outlined within
the wiki, noting that with the exception of your home directory, data on g2
filesystems is not backed up.

Home directories are shared via NFS and have a strict 10GB quota. All large
data files, such as job output, should be written to your /lustre/projects/
area. You can use the 'quota' command to check usage on both filesystems.
-----
The g2 head-node should only be used as an entry point to the supercomputer
and for submitting jobs to the queue nodes. Jobs found to be running directly
on the head-node may be removed without notice if affecting performance. We
have a set of interactive nodes - gstar001, gstar002, sstar001 and sstar002 -
that can be used to run short jobs directly, for testing code, etc.
*****
[guest01@g2 ~]$ pwd
/home/guest01
[guest01@g2 ~]$ |
```

INTRODUCTION TO gSTAR – ENVIRONMENT MODULES

Environment modules allow users to dynamically modify their environment.

Advantages include:

- *Handling of multiple software versions*
- *Handling of software dependencies*
- *Use of software installations that are optimised for the system*
- *Ease of use and reliability*

```
[g2: ~] > module avail openmpi
----- /usr/local/modules/modulefiles -----
openmpi/x86_64(default)
openmpi/x86_64/gnu/1.10.2-psm
openmpi/x86_64/gnu/1.4.4
openmpi/x86_64/gnu/1.4.5
openmpi/x86_64/gnu/1.6
openmpi/x86_64/gnu/1.6.1
openmpi/x86_64/gnu/1.6.1-nopbs
openmpi/x86_64/gnu/1.6.1-psm(default)
openmpi/x86_64/gnu/1.6.1-psm-nopbs
openmpi/x86_64/gnu/1.6.4
openmpi/x86_64/gnu/1.8
openmpi/x86_64/gnu/1.8.1-psm
openmpi/x86_64/gnu/1.8.3
openmpi/x86_64/gnu/1.8.3-dlopen
openmpi/x86_64/intel(default)
openmpi/x86_64/intel/1.10.2-psm
openmpi/x86_64/intel/1.6.4-psm(default)
openmpi/x86_64/intel/1.8.1-psm
openmpi/x86_64/intel/1.8.3-psm
openmpi/x86_64/intel/1.8.6-psm
[g2: ~] >
```

INTRODUCTION TO gSTAR – ENVIRONMENT MODULES

Environment module commands take the form:

`module <command> [arguments]`

To get a list of packages available on the system, use the 'avail' command:

`module avail`

To get a list of all the versions available for just one package, specify its name:

`module avail openmpi`

```
[g2: ~] > module avail openmpi
----- /usr/local/modules/modulefiles -----
openmpi/x86_64(default)
openmpi/x86_64/gnu/1.10.2-psm
openmpi/x86_64/gnu/1.4.4
openmpi/x86_64/gnu/1.4.5
openmpi/x86_64/gnu/1.6
openmpi/x86_64/gnu/1.6.1
openmpi/x86_64/gnu/1.6.1-nopbs
openmpi/x86_64/gnu/1.6.1-psm(default)
openmpi/x86_64/gnu/1.6.1-psm-nopbs
openmpi/x86_64/gnu/1.6.4
openmpi/x86_64/gnu/1.8
openmpi/x86_64/gnu/1.8.1-psm
openmpi/x86_64/gnu/1.8.3
openmpi/x86_64/gnu/1.8.3-dlopen
openmpi/x86_64/intel(default)
openmpi/x86_64/intel/1.10.2-psm
openmpi/x86_64/intel/1.6.4-psm(default)
openmpi/x86_64/intel/1.8.1-psm
openmpi/x86_64/intel/1.8.3-psm
openmpi/x86_64/intel/1.8.6-psm
[g2: ~] >
```

INTRODUCTION TO gSTAR – ENVIRONMENT MODULES

To apply all the user environment changes needed to use a version of openmpi compiled with the gnu compiler (for example), 'load' it:

```
module load openmpi/x86_64/gnu/1.4.4
```

Every module has a default version (usually the most up-to-date). To always load the default version of openmpi (for example), simply:

```
module load openmpi
```

To cleanly undo the effects of loading a module, 'unload' it. For example:

```
module unload openmpi
```

INTRODUCTION TO gSTAR – ENVIRONMENT MODULES

Use the following commands to navigate & use gSTAR's environment modules:

Command	Action
module avail	Display all modules available on the system
module avail [modulefile]	Display all available versions for a specific module
module list	Display all modules currently installed
module show [modulefile]	Display exactly what loading a specific module will do
module load [modulefile]	Load a specific module
module unload [modulefile]	Unload a specific module
module purge	Unload all modules

INTRODUCTION TO gSTAR – ENVIRONMENT MODULES

Insert demonstration here

```
karma/x86_64/1.7.25
kdtree/x86_64/gnu/latest
kepler/2.4(default)
lammps/x86_64/gnu/05-02-14
lammps/x86_64/gnu/13-01-14
lammps/x86_64/gnu/201505
lammps/x86_64/gnu/23Mar16
lammps/x86_64/gnu/30Jul16
lammps/x86_64/gnu/test
lammps/x86_64/intel/13-01-14
lammps/x86_64/intel/26-02-14
lammps/x86_64/intel/28-07-14
lapack/x86_64/gnu/3.2.1-4
lapack/x86_64/gnu/3.5.0
libctl/x86_64/gnu/3.2.1
libhugetlbfs/x86_64/gnu/2.14
libjpegturbo/x86_64/gnu/1.4.1
libtool/x86_64/gnu/2.4
lnr/unknown(default)
vmd/x86_64/1.9.1
vncviewer/x86_64/5.0.6
voro++/x86_64/gnu/0.4.6
vpfit/x86_64/gnu/10
vpfit/x86_64/gnu/10.2
vtk/x86_64/gnu/5.10.1
vtk/x86_64/gnu/6.3.0
wcslib/x86_64/gnu/4.17
wcstools/x86_64/gnu/3.9.0
wxwidgets/3.1.0
xaw3d/x86_64/gnu/1.5
xdvi/x86_64/gnu/22.87
xemacs/x86_64/gnu/21.5.33
xftdtd/7.6.0.5
xidl/x86_64/gnu/r3326
xz/x86_64/5.2.2
yorick/x86_64/gnu/2.1
yorick/x86_64/gnu/2.1.06

[guest01@g2 ~]$ module load openmpi/x86_64/gnu/1.4.4
[guest01@g2 ~]$ module list
Currently Loaded Modulefiles:
  1) openmpi/x86_64/gnu/1.4.4
[guest01@g2 ~]$ module unload openmpi
[guest01@g2 ~]$ module list
No Modulefiles Currently Loaded.
[guest01@g2 ~]$ module load openmpi
[guest01@g2 ~]$ module list
Currently Loaded Modulefiles:
  1) intel/12.1.0
[guest01@g2 ~]$
```

2) openmpi/x86_64/intel/1.6.4-psm

INTRODUCTION TO gSTAR – QUOTAS

Every user may belong to their own unique set of projects. Project membership and the storage quotas associated with each can be queried:

```
[guest01@g2 ~]$ pwd
/home/guest01
[guest01@g2 ~]$ groups
p093_swin
[guest01@g2 ~]$ cd /projects/p093_swin/
[guest01@g2 p093_swin]$ quota
Disk quotas for user guest01 (uid 1047):
  Filesystem blocks  quota  limit  grace  files  quota  limit  grace
/home
          268 10485760 10485760
          77      0      0      (Disk usage is at 0.0%, 268kB of 10.0GB)
Disk quotas for group p093_swin (gid 10120): none
Disk quotas for group p093_swin (gid 10120):
  Filesystem kbytes  quota  limit  grace  files  quota  limit  grace
/projects      28 10737418240 11811160064 -      8 1000000 1050000  -(Disk usage is at 0.0%, 28kB of 10240.
0GB, Inode usage is at 0.0%)
[guest01@g2 p093_swin]$ user-quota
Disk quotas for group p093_swin (gid 10120):
  Filesystem kbytes  quota  limit  grace  files  quota  limit  grace
/projects      28 10737418240 11811160064 -      8 1000000 1050000  -(Disk usage is at 0.0%, 28kB of 10240.
0GB, Inode usage is at 0.0%)
[guest01@g2 p093_swin]$ |
```

INTRODUCTION TO gSTAR – QUOTAS

Insert demonstration here

```
[guest01@g2 ~]$ pwd
/home/guest01
[guest01@g2 ~]$ groups
p093_swin
[guest01@g2 ~]$ cd /projects/p093_swin/
[guest01@g2 p093_swin]$ quota
Disk quotas for user guest01 (uid 1047):
  Filesystem blocks  quota  limit  grace  files  quota  limit  grace
/home
          268 10485760 10485760          77      0      0      (Disk usage is at 0.0%, 268kB of 10.0GB)
Disk quotas for group p093_swin (gid 10120): none
Disk quotas for group p093_swin (gid 10120):
  Filesystem kbytes  quota  limit  grace  files  quota  limit  grace
/projects      28 10737418240 11811160064  -      8 1000000 1050000  -(Disk usage is at 0.0%, 28kB of 10240.
0GB, Inode usage is at 0.0%)
[guest01@g2 p093_swin]$ user-quota
Disk quotas for group p093_swin (gid 10120):
  Filesystem kbytes  quota  limit  grace  files  quota  limit  grace
/projects      28 10737418240 11811160064  -      8 1000000 1050000  -(Disk usage is at 0.0%, 28kB of 10240.
0GB, Inode usage is at 0.0%)
[guest01@g2 p093_swin]$ |
```

INTRODUCTION TO gSTAR – QUOTAS

Some-or-all of the following quota values may be defined for each filesystem:

Value	Description
filesystem	➤ The mounted file system, usually in the form <server>:/<directory>. ➤ Use df <directory> to determine the mounted file system for a given directory.
blocks	➤ Used disk space in kilobytes.
quota	➤ Soft limit for disk usage. ➤ Writing to the filesystem is still permitted, but only until the grace time is up.
limit	➤ Hard limit for disk usage. ➤ Writing new data is completely blocked when exceeded.
grace	➤ The time left to remove data, when the soft quota is exceeded. ➤ When this time is up, writing of new data is blocked until quota excess is cleared.
file	➤ Number of files.
quota	➤ Soft limit for the number of files. ➤ Writing new files is still permitted, but only until the grace time is up).
limit	➤ Hard limit for number of files. ➤ Writing new files is completely blocked when exceeded.
grace	➤ The time left to remove files, when the soft file quota is exceeded. ➤ When this time is up, new file creation is blocked until quota excess is cleared.

INTRODUCTION TO gSTAR – THE PORTABLE BATCH SYSTEM (PBS)

To execute a job on the compute nodes, a user must first create a submission script and place PBS directives at the beginning to specify needed resources, etc.

See the following example located at:

`/projects/p093_swin/samplePBS.sh`

```
#!/bin/bash -f
```

```
#PBS -N TestJob
```

```
# By default, the standard output and error streams are sent to files in the current working directory with names:
```

```
# job_name.osequence_number <- output stream
```

```
# job_name.esquence_number <- error stream
```

```
# where job_name is the name of the job and sequence_number is the job
```

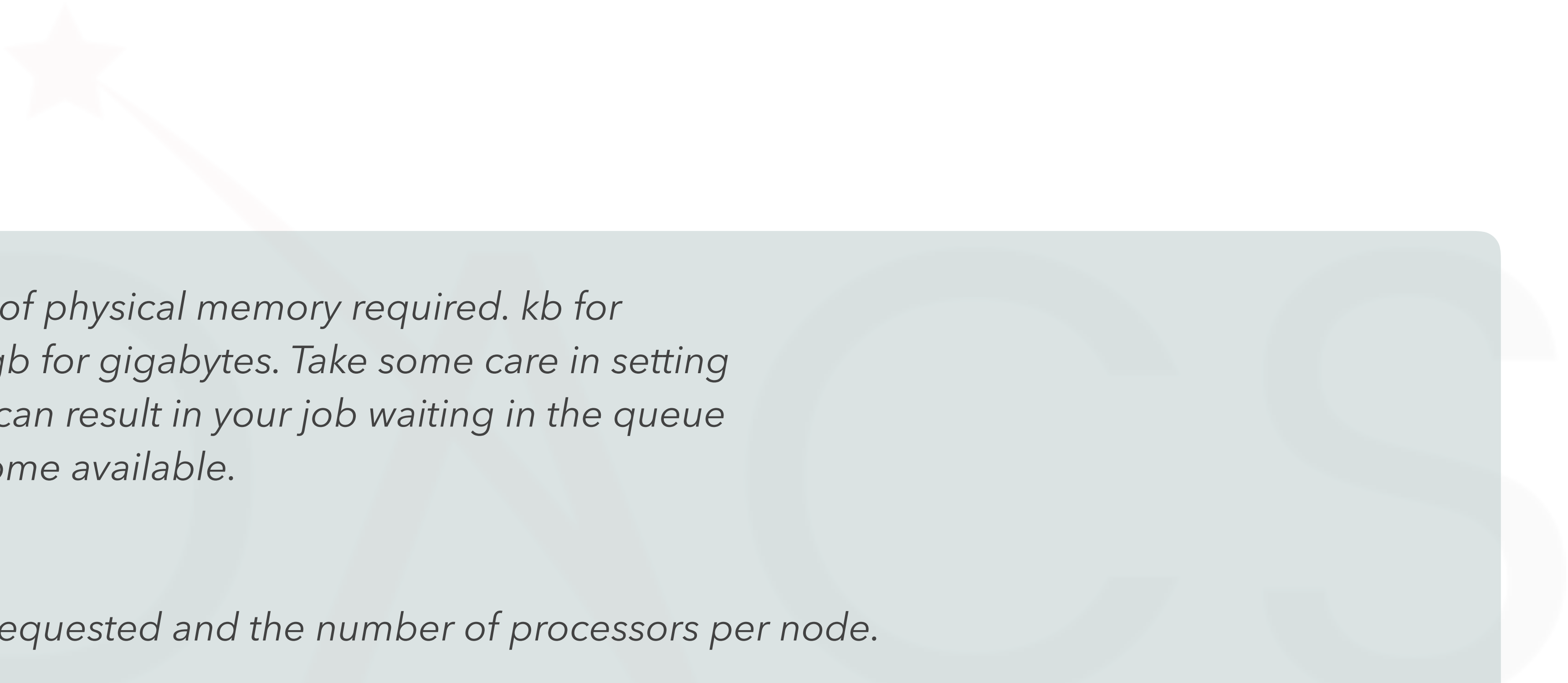
```
# number assigned when the job is submitted. Use the directives below to
```

```
# change the files to which the standard output and error streams are sent.
```

```
#PBS -o stdout_file
```

```
#PBS -e stderr_file
```

INTRODUCTION TO gSTAR – THE PORTABLE BATCH SYSTEM (PBS)



*# Specify the maximum amount of physical memory required. kb for
kilobytes, mb for megabytes, gb for gigabytes. Take some care in setting
this value. Setting it too large can result in your job waiting in the queue
for sufficient resources to become available.*

#PBS -l mem=512mb

Specify the number of nodes requested and the number of processors per node.

#PBS -l nodes=1:ppn=1

ASTRONOMY DATA AND COMPUTING SERVICES

INTRODUCTION TO gSTAR – THE PORTABLE BATCH SYSTEM (PBS)

To get a summary of the PBS resources supported by gSTAR, run the following:

```
man pbs_resources
```

Each resource can be set as a directive in the submit script, or as an option to qsub:

Resource	qsub option	Script Directive (+ example)
Wallclock time	-l walltime=hh:mm:ss	#PBS -l walltime=10:00:00
Nodes/Processors-per-node	-l nodes=n:ppn=p	#PBS -l nodes=2:ppn=12
... + GPUs	-l nodes=n:ppn=p:gpus=g	#PBS -l nodes=1:ppn=12:gpus=2
Memory per processor	-l pmem=SIZEgb	#PBS -l pmem=2gb
Name of the job	-N jobname	#PBS -N Test_job2
Queue to use	-q queueName	#PBS -q gstar
Project to use	-A projectID	#PBS -A p093_swin

INTRODUCTION TO gSTAR – THE PORTABLE BATCH SYSTEM (PBS)

The following commands can be used to create, destroy or get the status of jobs on the queue:

Action	PBS command
Submit a job	qsub jobscript
Delete a job	qdel jobID
Delete all jobs of user	qdel 'qselect -u user'
Status of all jobs of user	qstat -u user
Status of all jobs	qstat -a
Status of a specific job	qstat [-f] jobID

INTRODUCTION TO gSTAR – THE PORTABLE BATCH SYSTEM (PBS)

Insert demonstration here

```
[g2: p093_swin] > pwd
/projects/p093_swin
[g2: p093_swin] > ls
samplePBS.sh
[g2: p093_swin] > qsub samplePBS.sh
5503433.pbs.hpc.swin.edu.au
[g2: p093_swin] > ls
samplePBS.sh TestJob.e5503433 TestJob.o5503433
[g2: p093_swin] > more TestJob.o5503433
-----
Job is running on node gstar046
-----
PBS: qsub is running on g2.hpc.swin.edu.au
PBS: originating queue is gstar
PBS: executing queue is gstar
PBS: working directory is /projects/p093_swin
PBS: execution mode is PBS_BATCH
PBS: job identifier is 5503433.pbs.hpc.swin.edu.au
PBS: job name is TestJob
PBS: current home directory is /home/guest01
PBS: PATH = /usr/lib64/qt-3.3/bin:/bin:/usr/bin:/usr/X11R6/bin:/opt/torque/bin:/opt/mpi/bin:/opt/sgl/
sgimc/bin:/home/guest01/bin
-----
[g2: p093_swin] > more TestJob.e5503433
[g2: p093_swin] > |
```

INTRODUCTION TO gSTAR – THE PORTABLE BATCH SYSTEM (PBS)

In addition to the 'qstat' command, the 'showq' command presents a good summary of the jobs running or queued on the system:

5501313	jzhang	Running	16	1:01:41:58	Fri Jun 9	00:24:07
5501311	jzhang	Running	16	1:01:47:08	Fri Jun 9	00:29:17
5501315	jzhang	Running	16	1:01:47:21	Fri Jun 9	00:29:30
5503524	cward	Running	1	00:22:44	Fri Jun 9	00:34:53
5503525	cward	Running	1	00:25:17	Fri Jun 9	00:37:26
5503527	cward	Running	1	00:26:05	Fri Jun 9	00:38:14
5503528	cward	Running	1	00:26:16	Fri Jun 9	00:38:25
5503529	cward	Running	1	00:26:49	Fri Jun 9	00:38:58
5503530	cward	Running	1	00:26:58	Fri Jun 9	00:39:07
5503531	cward	Running	1	00:27:01	Fri Jun 9	00:39:10
5503532	cward	Running	1	00:27:25	Fri Jun 9	00:39:34
5503533	cward	Running	1	00:27:34	Fri Jun 9	00:39:43
219 active jobs		1576 of 2120 processors in use by local jobs (74.34%)				
		123 of 143 nodes active (86.01%)				
eligible jobs-----						
JOBID	USERNAME	STATE	PROCS	WCLIMIT	QUEUE TIME	
5429234	tnose	Idle	160	7:00:00:00	Thu May 25	17:12:40
5429238	tnose	Idle	160	7:00:00:00	Thu May 25	17:14:34
5429289	tnose	Idle	160	7:00:00:00	Thu May 25	17:16:07
5479395	cperry	Idle	192	5:00:00	Tue Jun 6	12:36:58
5449205	skannam	Idle	16	7:00:00:00	Thu Jun 1	11:20:45
5449196	skannam	Idle	16	7:00:00:00	Thu Jun 1	11:20:45
5449194	skannam	Idle	16	7:00:00:00	Thu Jun 1	11:20:44
5449204	skannam	Idle	16	7:00:00:00	Thu Jun 1	11:20:45

INTRODUCTION TO gSTAR – THE PORTABLE BATCH SYSTEM (PBS)

Code can also be run on the compute nodes interactively by starting an "interactive session":

```
[g2: ~] > qsub -I -l nodes=1:ppn=1 -l pmem=512mb
qsub: waiting for job 5503428.pbs.hpc.swin.edu.au to start
qsub: job 5503428.pbs.hpc.swin.edu.au ready

[gstar046: ~] > hostname
gstar046
[gstar046: ~] > echo $PBS_JOBID $PBS_ENVIRONMENT $PBS_WALLTIME
5503428.pbs.hpc.swin.edu.au PBS_INTERACTIVE 900
[gstar046: ~] > exit
logout

qsub: job 5503428.pbs.hpc.swin.edu.au completed
[g2: ~] > |
```

INTRODUCTION TO gSTAR – THE PORTABLE BATCH SYSTEM (PBS)

Insert demonstration here

```
[guest01@g2 ~]$ qsub -I -l nodes=1:ppn=1 -l pmem=512mb  
qsub: waiting for job 5520108.pbs.hpc.swin.edu.au to start  
qsub: job 5520108.pbs.hpc.swin.edu.au ready
```

```
[guest01@gstar057 ~]$ hostname  
gstar057
```

```
[guest01@gstar057 ~]$ echo $PBS_JOBID  
5520108.pbs.hpc.swin.edu.au
```

```
[guest01@gstar057 ~]$ echo $PBS_ENVIRONMENT  
PBS_INTERACTIVE
```

```
[guest01@gstar057 ~]$ echo $PBS_WALLTIME  
900
```

```
[guest01@gstar057 ~]$ exit  
logout
```

```
qsub: job 5520108.pbs.hpc.swin.edu.au completed  
[guest01@g2 ~]$
```

INTRODUCTION TO gSTAR – THE PORTABLE BATCH SYSTEM (PBS)

If you are a member of multiple projects, don't forget to specify which project you want computing time assigned to when you create a new job:

`qsub -A projectID`

or

`#PBS -A projectID`

WORKING WITH QUEUES – THE GANGLIA MONITORING SYSTEM

The Ganglia system is available at: <http://g2.hpc.swin.edu.au/ganglia/>

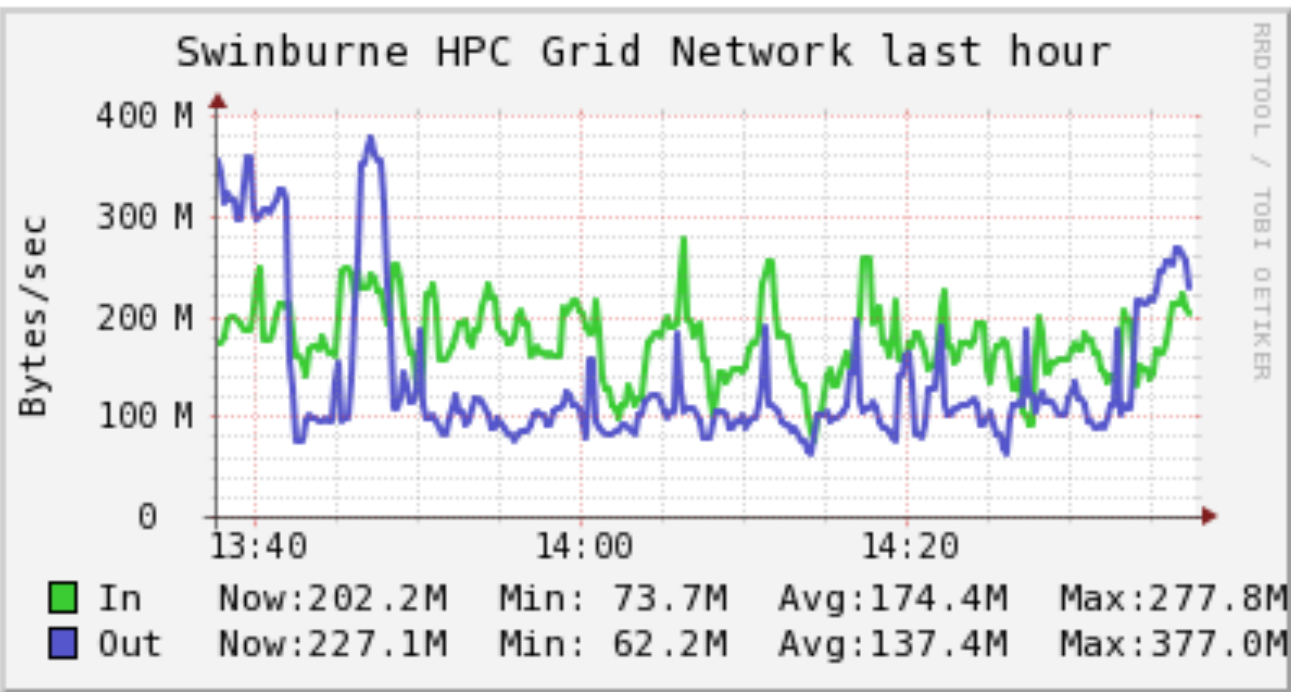
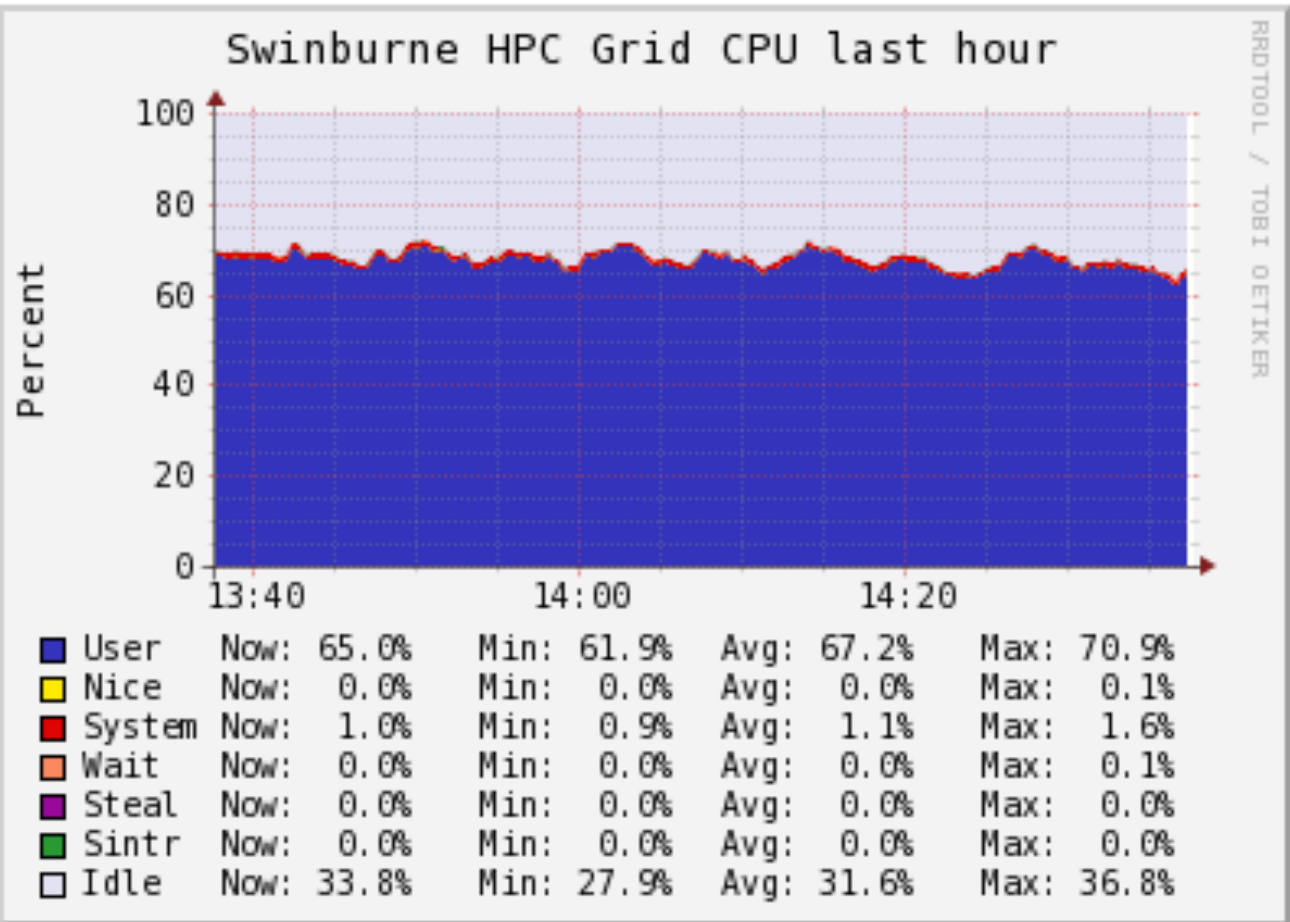
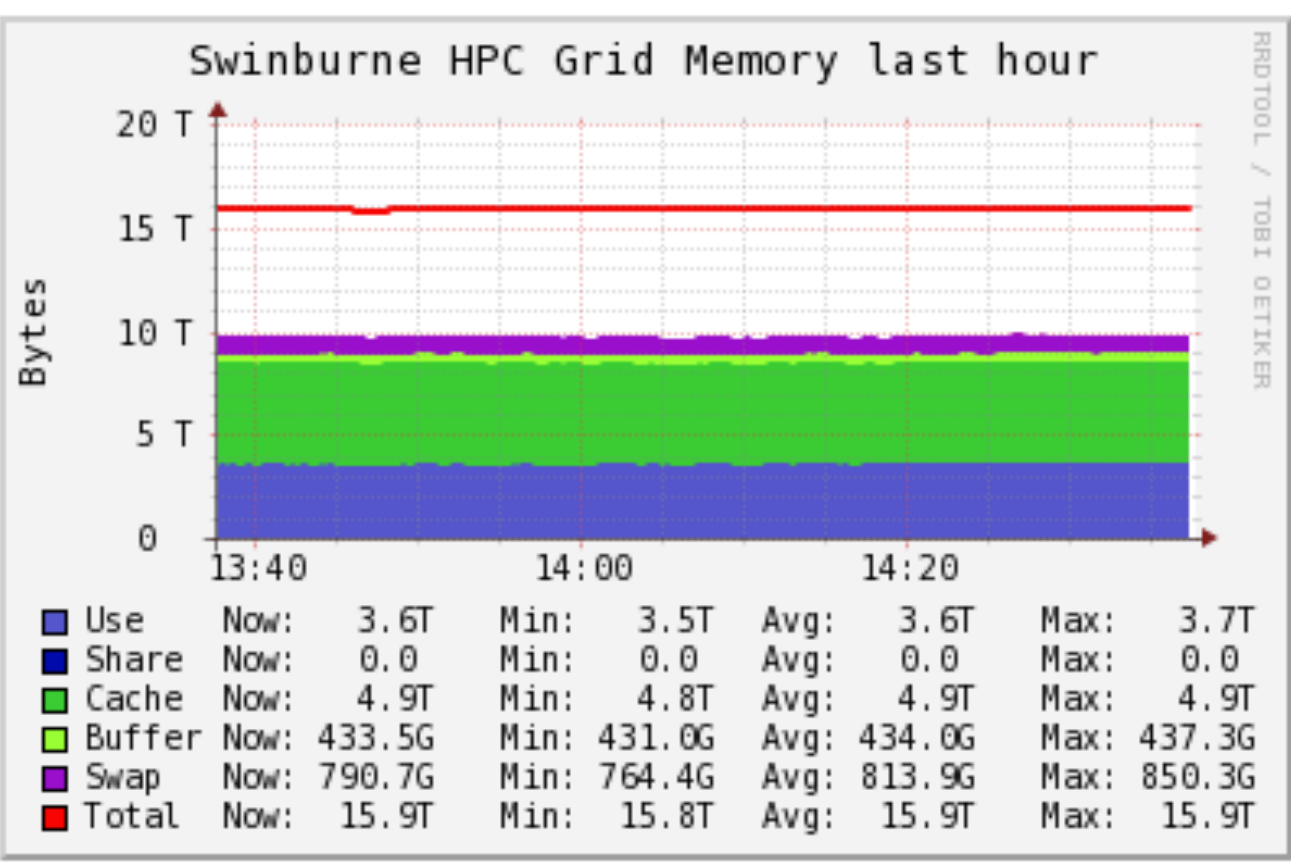
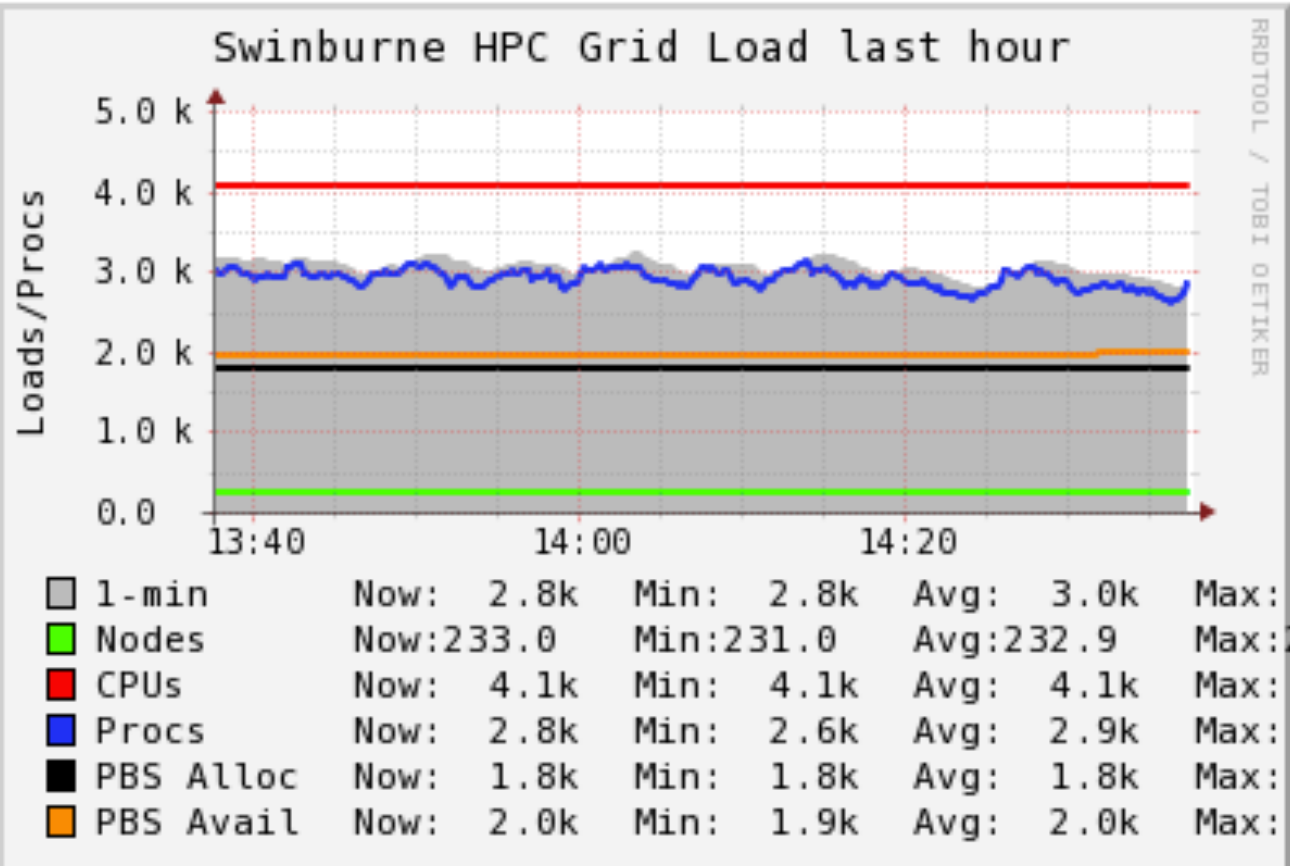
Swinburne HPC Grid (3 sources) (tree view)

CPU's Total: **4080**
Hosts up: **233**
Hosts down: **1**

Current Load Avg (15, 5, 1m):
73%, 71%, 69%

Avg Utilization (last hour):
75%

Localtime:
2017-06-05 14:37



WORKING WITH QUEUES – THE GANGLIA MONITORING SYSTEM

Ganglia presents a broad summary of the cluster's activity ...

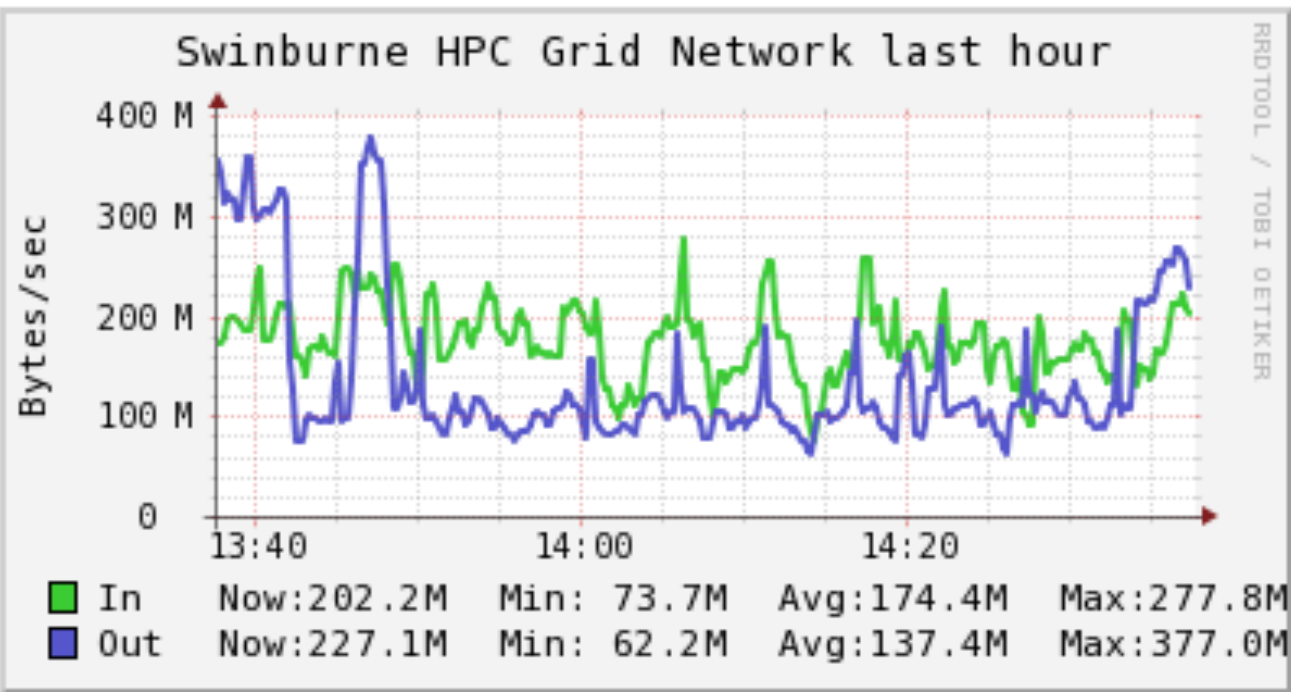
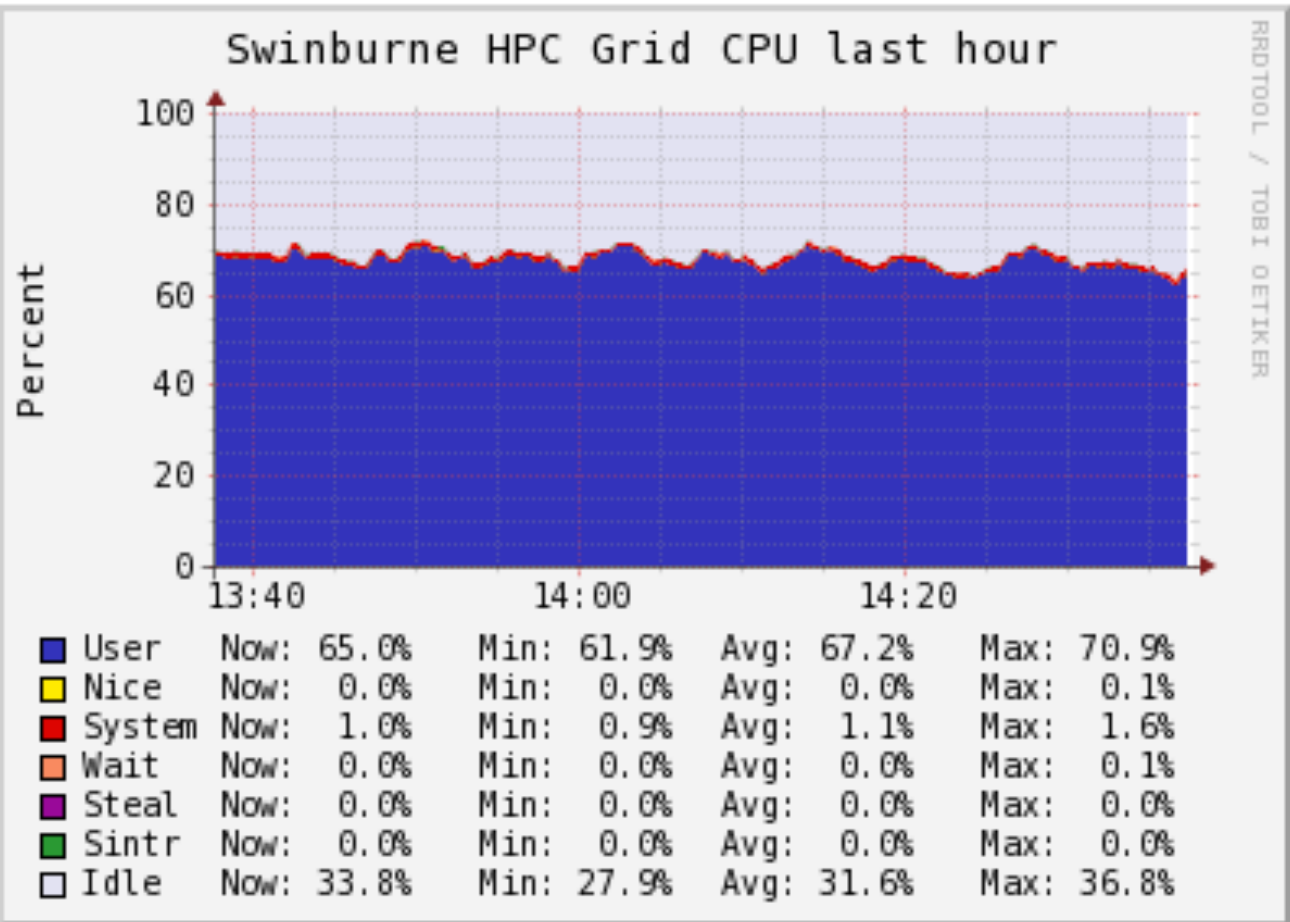
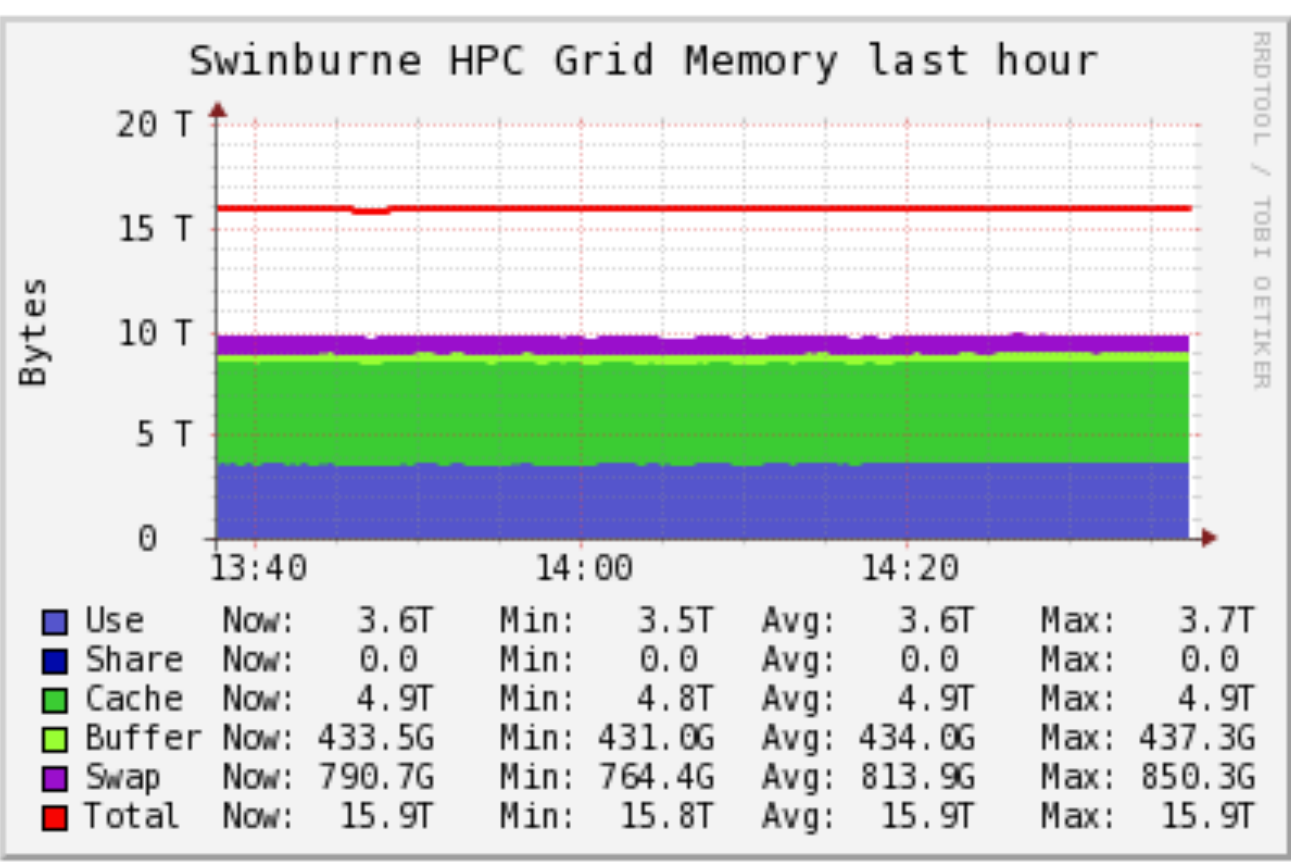
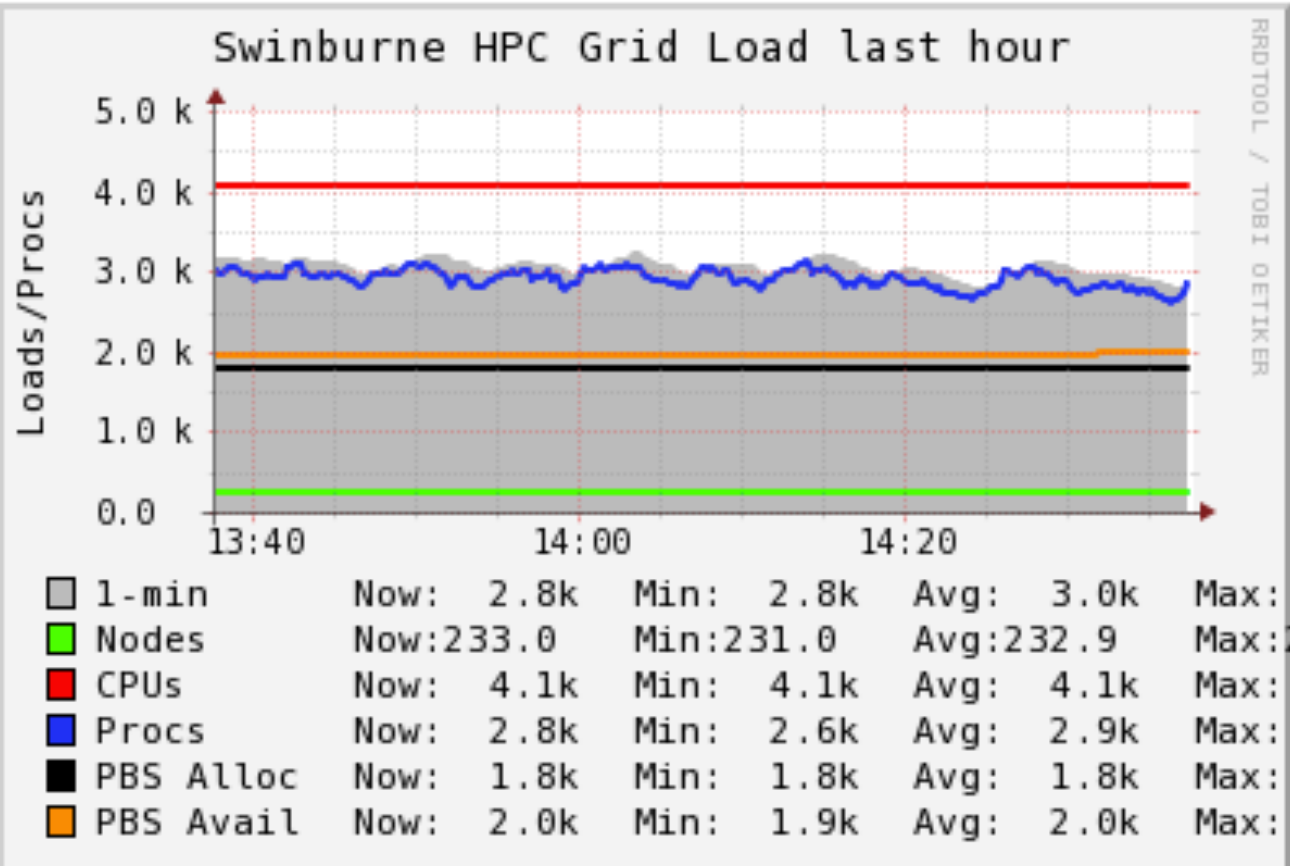
Swinburne HPC Grid (3 sources) (tree view)

CPU's Total: **4080**
Hosts up: **233**
Hosts down: **1**

Current Load Avg (15, 5, 1m):
73%, 71%, 69%

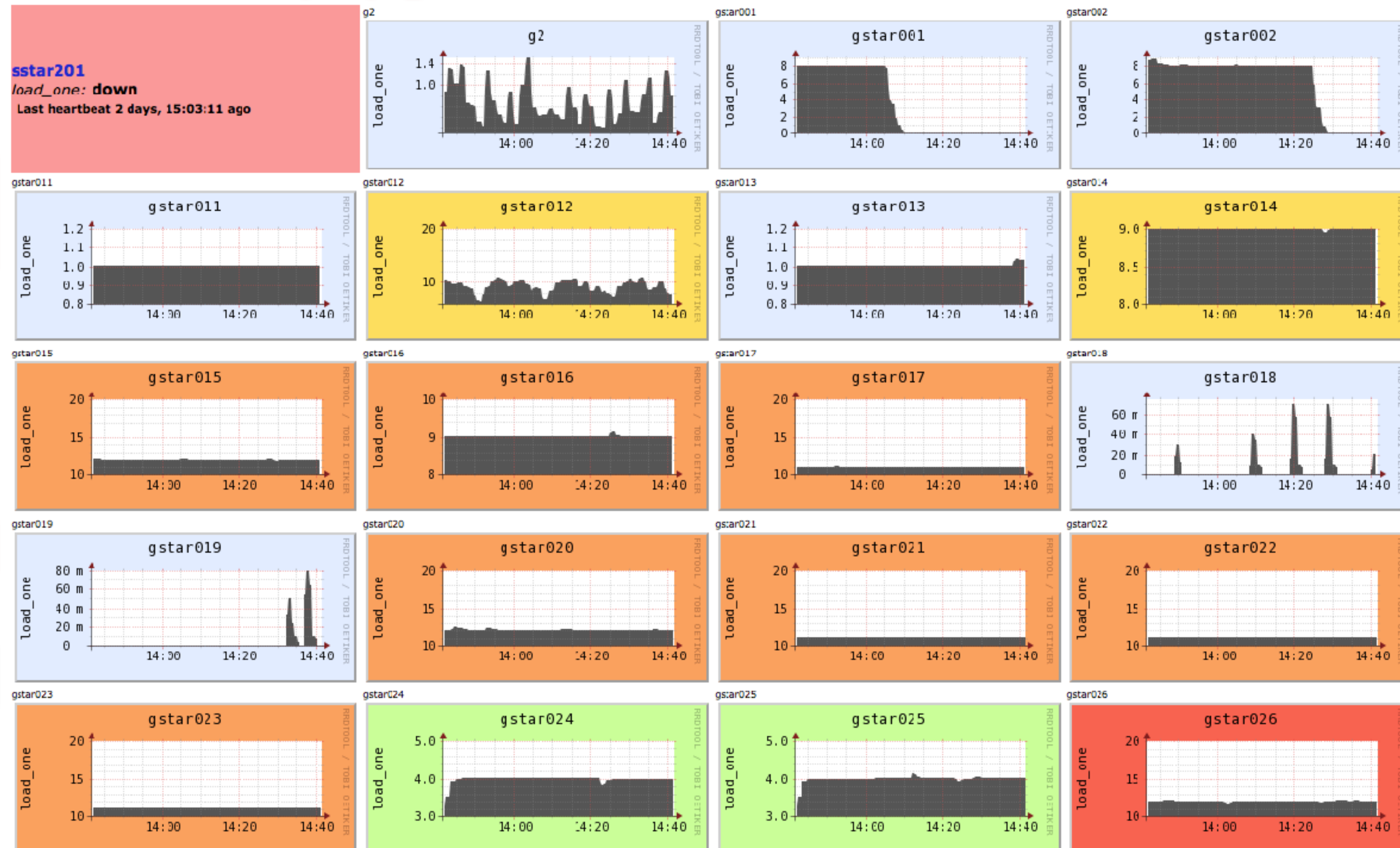
Avg Utilization (last hour):
75%

Localtime:
2017-06-05 14:37



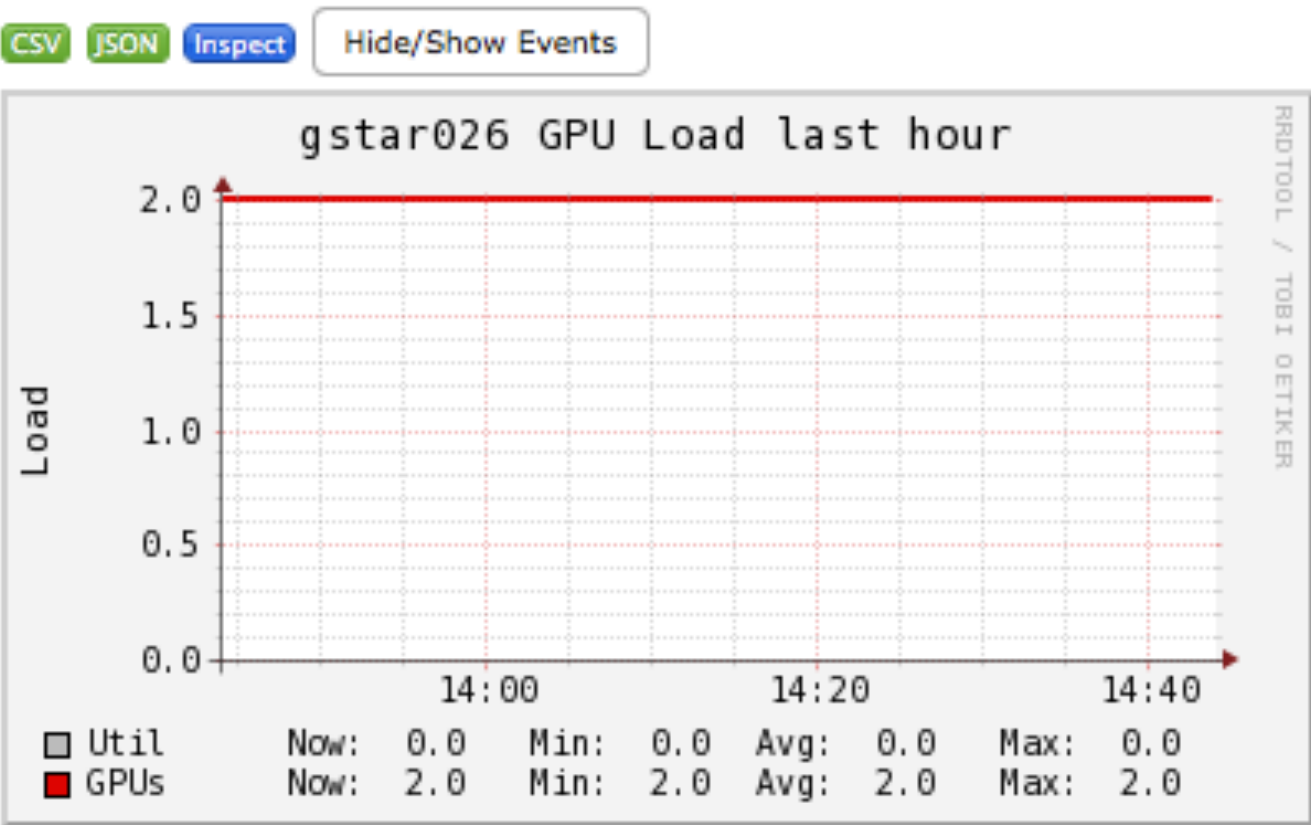
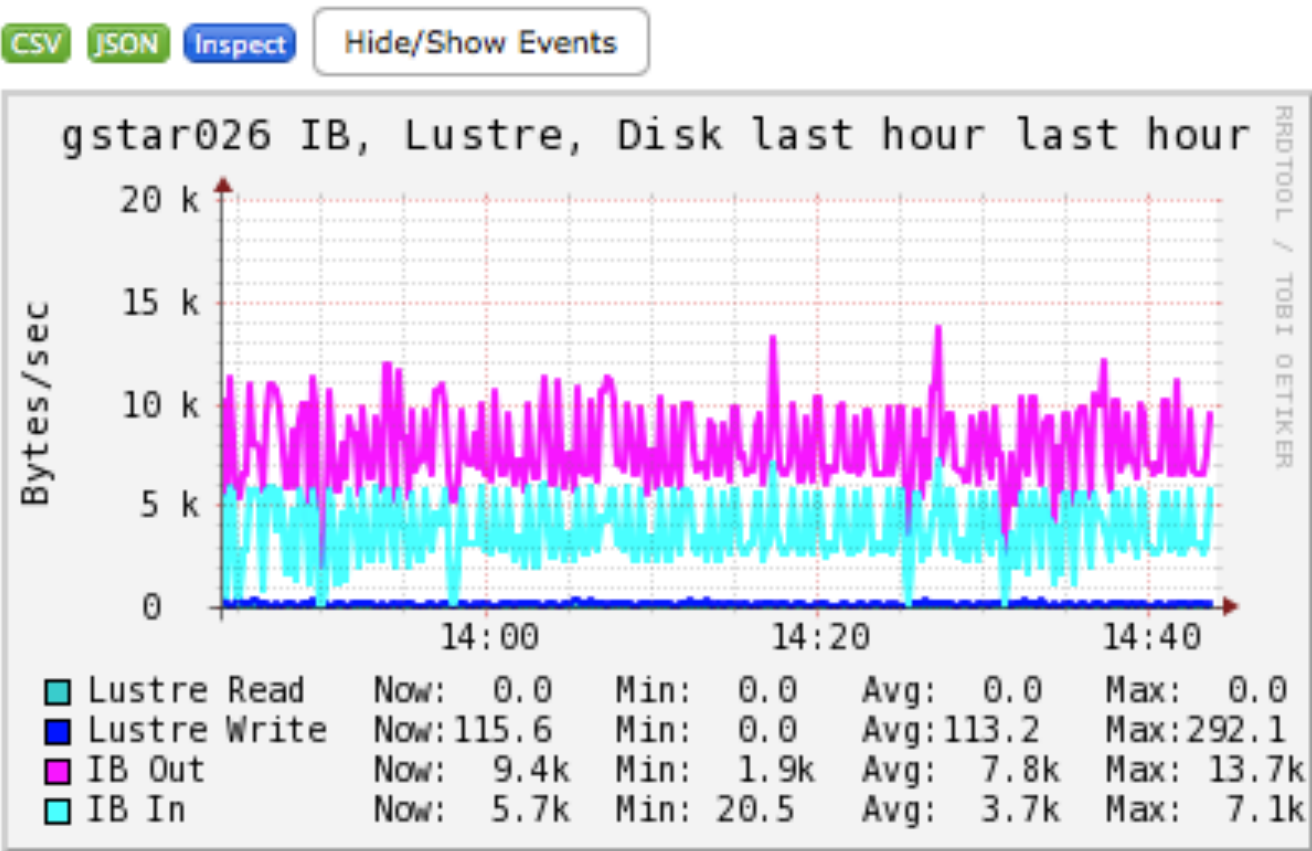
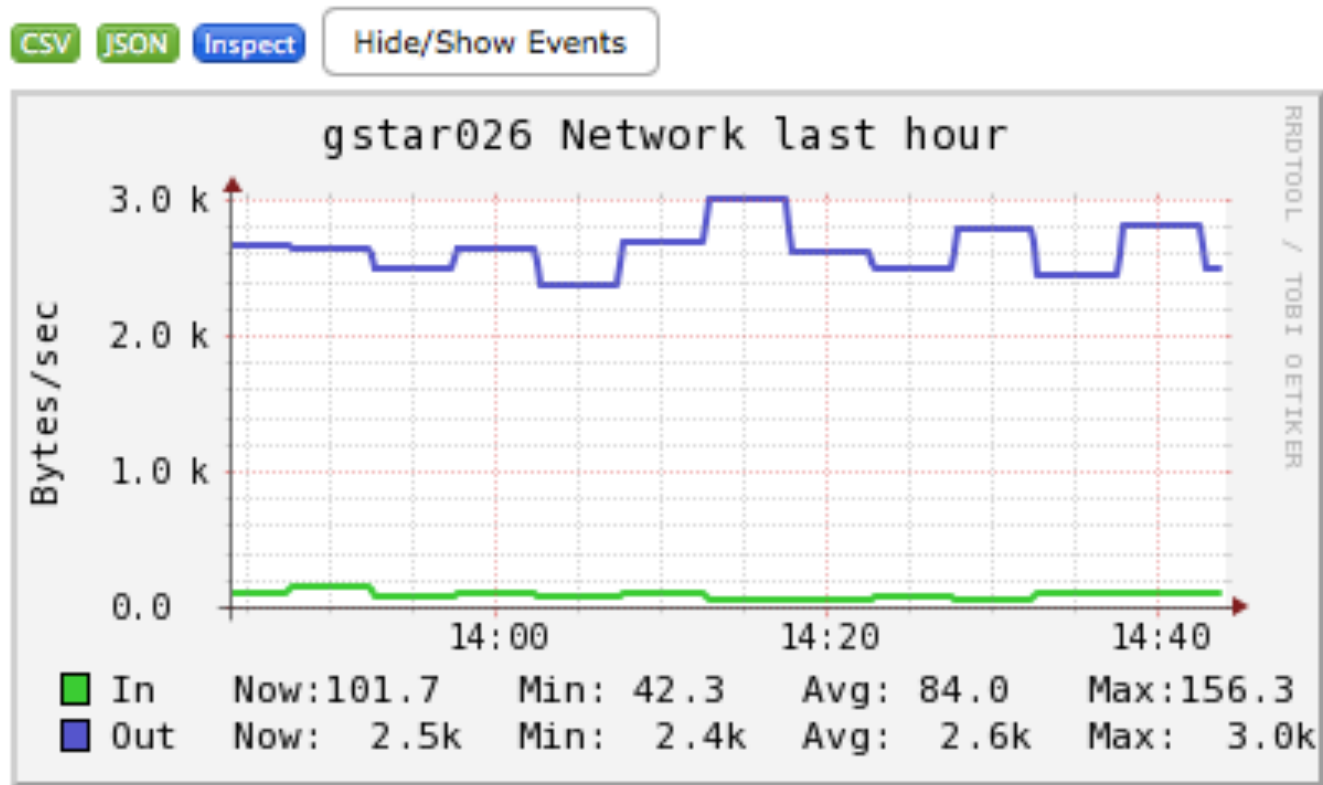
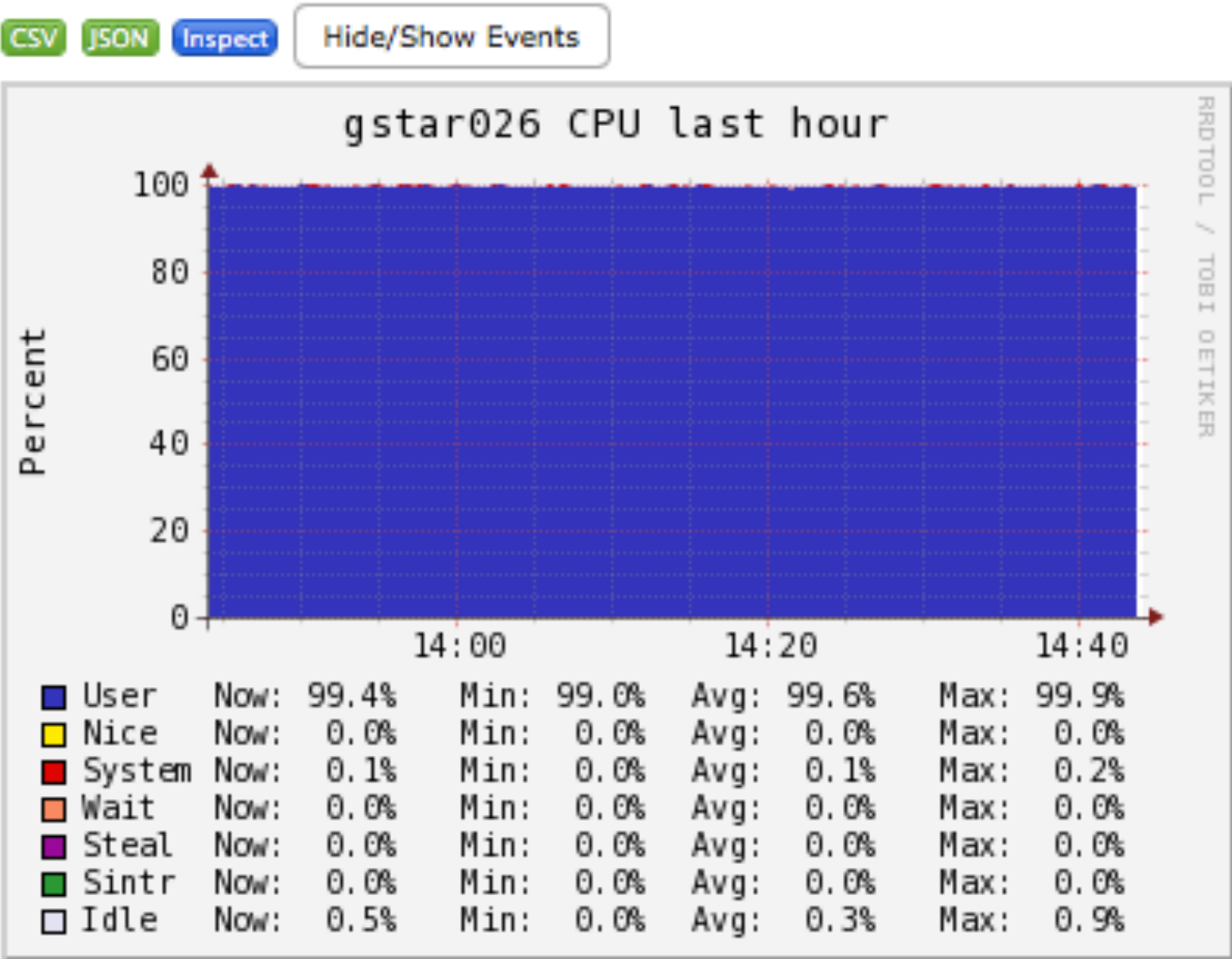
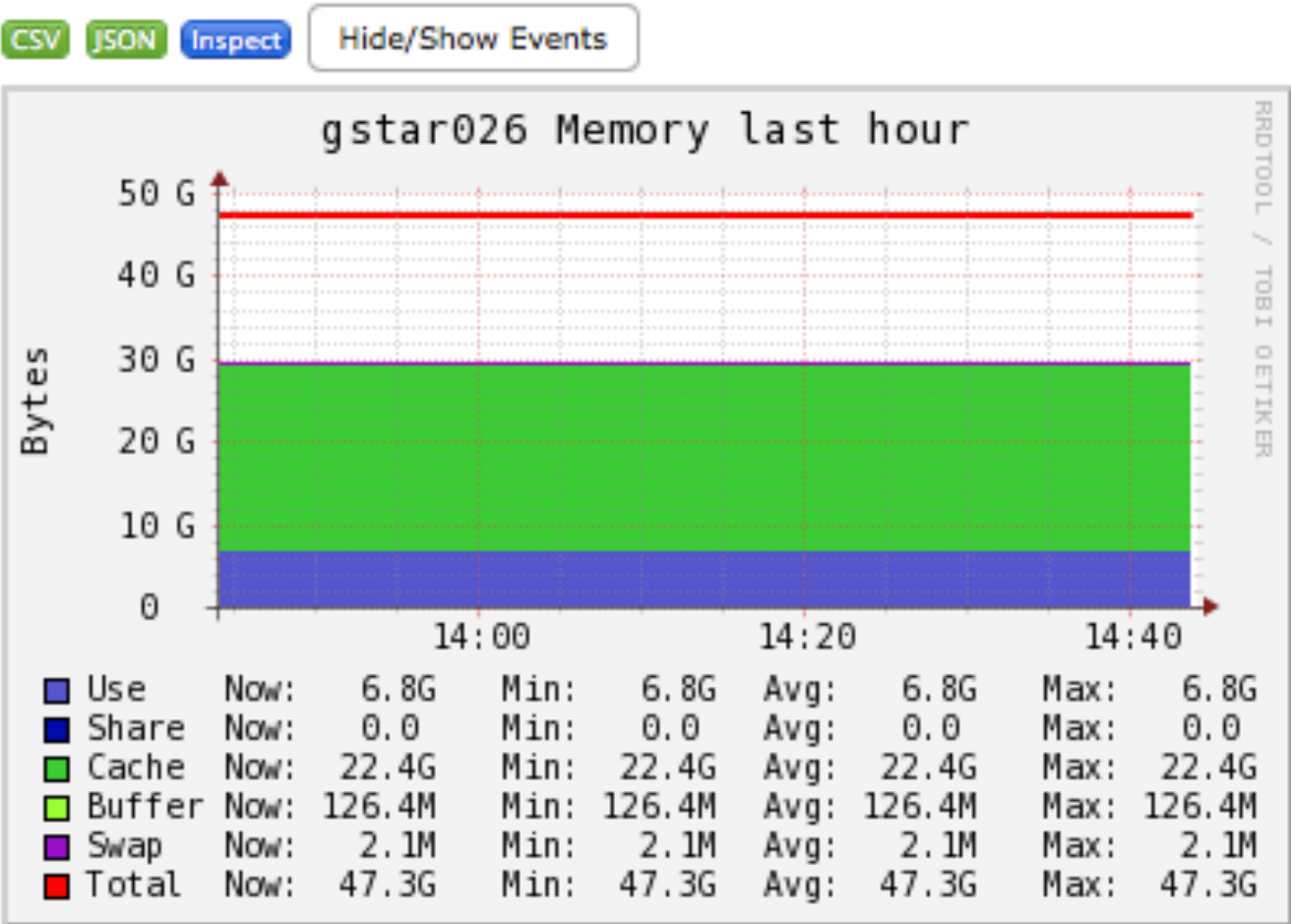
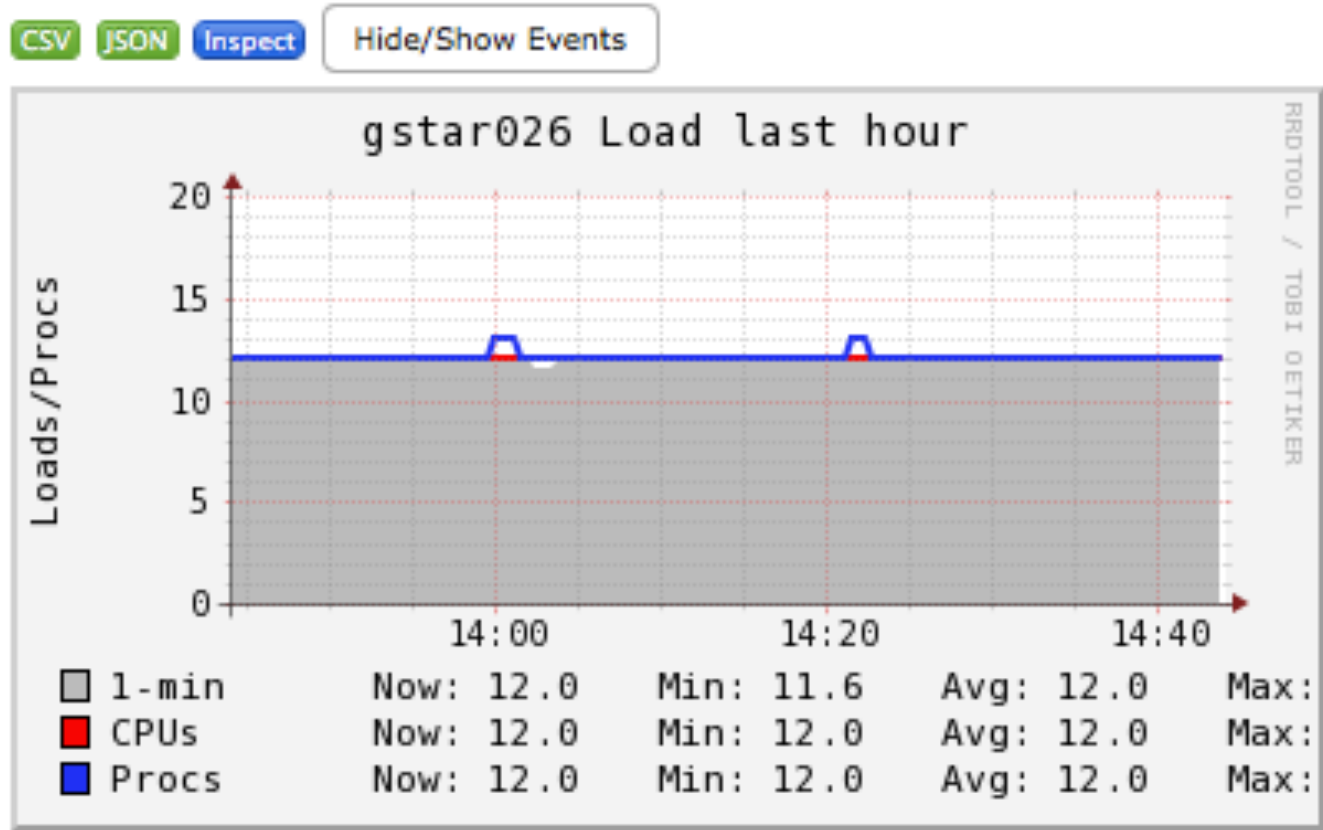
WORKING WITH QUEUES – THE GANGLIA MONITORING SYSTEM

... a course-grained summary of all the cluster's nodes ...



WORKING WITH QUEUES – THE GANGLIA MONITORING SYSTEM

... and a fine-grained summary detailing the activity of each cluster node ...



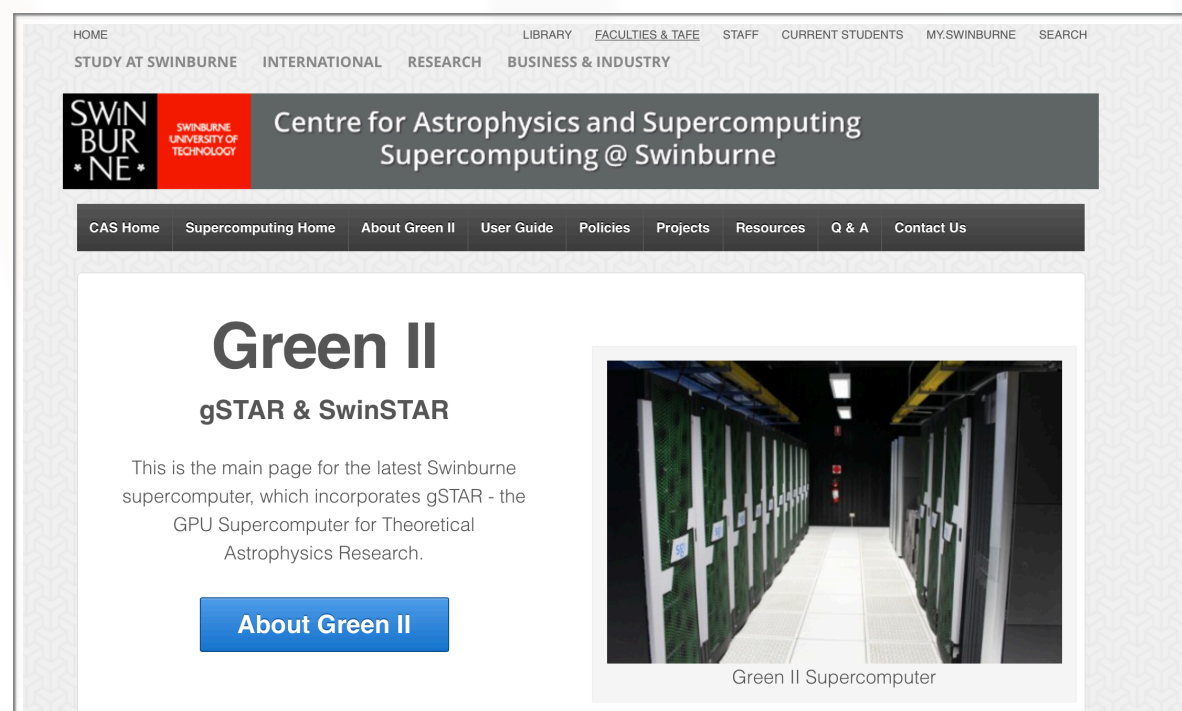
INTRODUCTION TO THE GSTAR CLUSTER AT SWINBURNE – MORE HELP

Reiterating: more help or information can be obtained from the following:



... email the Swinburne HPC Help Desk:

HPC-Support@swin.edu.au



... or visit the gSTAR webpage:

supercomputing.swin.edu.au