# Using The Batch System: Univa Grid Engine (UGE) at The JGI

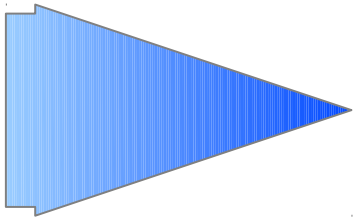
JGI Training
February 10, 2012

Ilya Malinov, Katie Antypas, Jay Srinivasan

U.S. DEPARTMENT OF ENERGY | Office of Science
BERKELEY LAB
Lawrence Berkeley National Laboratory
NERSC
National Energy Research Scientific Computing Center
JGI DOE JOINT GENOME INSTITUTE
US DEPARTMENT OF ENERGY
OFFICE OF SCIENCE

# Consolidated Computation Cluster

HYPERION
OCEANUS
CRIUS
IAPETUS
RHEA
THEIA
THEMIS
KRONOS



**CRIUS**

Benefits:

- Surge to higher core count
- Better utilization
- Easier maintenance

# CRIUS: HARDWARE

- 450 “commodity” nodes:
  - SGI, 8 CPU Cores, 48GB RAM
- 46 “commodity” nodes:
  - Supermicro, 8 CPU Cores, 48GB RAM
- 20 “high memory” nodes:
  - SUN, 8 CPU Cores, 144GB RAM
- 8 “high memory” nodes:
  - SGI, 24 CPU Cores, 254GB RAM
- 4 “high memory” nodes:
  - SUN, 32 CPU Cores, 512GB RAM
- 1 “high memory” node:
  - DELL, 32 CPU Cores, 1024GB RAM
- 1 “high memory” node:
  - DELL, 48 CPU Cores, 256GB RAM
- 2 “high memory nodes:
  - IBM, 1024GB RAM

# Using UGE (Univa Grid Engine)

- Current version: 8.0.1, http://www.univa.com
- Jobs run in queues
- Queues have instances on compute nodes
- Each queue instance has slots on compute nodes
- Currently the following queues are available:
  - bg.q
  - debug.q
  - normal.q
  - short.q
  - system.q
  - timelogic.q

- Source the environment: /opt/uge/crius/uge/crius/common/settings.(c)sh

# Queue Structure

| Queue Name | Purpose | Node Limit | Memory Limit | Wall Clock Limit | Job Limit | Slot Limit |
|---|---|---|---|---|---|---|
| debug.q | Fast turnaround for debugging purposes | 3 | 48GB | 8 h | 1 | 16 |
| normal.q | Production workflows | - | - | <= 12h* | - | - |
| | | 150** | - | > 12h | - | - |
| short.q | Short and light jobs | Comm. nodes | 8GB | 1h | - | 1/node |
| bg.q | Long, low priority jobs | - | 8GB | - | 100 | 200 |
| timelogic.q | Gives access to Timelogic accelerated blast nodes | Comm. nodes | - | - | - | 1/node |

* - Default is 12 hours.
** - 150 “commodity” nodes and all hight memory nodes.

# Requesting a Queue

Queues are requested via resources:

- bg.c --> bg.q
- debug.c --> debug.q
- normal.c --> normal.q
- short.c --> short.q
- timelogic.c --> timelogic.q

normal.q is now the default queue, *i.e.*, it does not to be explicitly requested.

The following resources will be deprecated: long.c, crius_bg.c, crius_high.c, crius_normal.c, galaxy_bg.c, galaxy_high.c, galaxy_normal.c, medium.c,

Example: `qsub -l debug.c <...>`

# Support for Parallel And Multi-CPU Jobs

Available parallel environments:

- pe_1
- pe_2
- pe_3
- pe_4
- pe_5
- pe_6
- pe_7
- pe_8
- pe_16
- pe_fill
- pe_rrobin
- pe_slots

Nomenclature:

- pe_<N>, where N is either an integer number of processes per host to use, or a special word:
- “fill” - all available slots on an node are allocated before dispatching to the next host.
- “rrobin” - a single slot per node is allocated on all available nodes. If more slots required, allocation starts with the first node again.
- “slots” - all processes will be allocated on a single node

Usage: `pe_<N> <xN>`, where xN – number of total processes requested.

Example: `qsub -pe pe_8 32 <...>`

# Array Jobs

- Array Job is an array of identical tasks being differentiated only by an index number. The index numbers are exported to the job tasks via the environment variable `SGE_TASK_ID`.
- Usage of array jobs is highly encouraged, as it allows to schedule hundreds of tasks with minimal load on the scheduler.
- To specify array jobs, a '`-t`' option should be passed to `qsub` in the form of `-t x[-y[:z]]`, *i.e.*, task index range may be a single number, a simple range of the form x-y or a range with a step size x-y:z.

Example: `qsub -t 2-10:2 <...>`

# Fair Share

- “Fair Share”, or Share-based scheduling insures each project receives its allocated share of CPU time over a period of time.
- Shares are adjusted for each scheduling interval.
- The default project for new users is jgi.p
- If a user is allowed to use more than one project, s/he can specify a desired project during job submission.

Example: `qsub -P gbp.p <...>`

| Project | Share (%) |
|---|---|
| annotation.p | 7.8 |
| comparative.p | 11.9 |
| gbp.p | 16.9 |
| img.p | 13.6 |
| jgi.p | 7.8 |
| mep.p | 1.7 |
| pi.p | 3.1 |
| plant.p | 27.1 |
| rnd.p | 10.1 |

# Other Useful Options for `qsub`

- `-r {y|n}` – indicates if this job should be re-scheduled in case of node crash. The default on crius' queues is 'yes'. If this job was re-scheduled, an environment variable `RESTARTED` is set.
- `-R {y|n}` – indicates whether a reservation for this job should be done. Default: 'no'.
- `-j {y|n}` – indicates whether to combine STDERR and STDOUT of this job in one file. Default: 'no'.
- `-b {y|n}` – indicates whether the job being submitted is a binary file. Default: 'no'.
- `-V` – specifies that all environment variables active within the qsub utility be exported to the context of the job.
- `-v <variable>[=value][,...]` – defines or redefines the environment variable(s) to be exported to the execution context of the job.
- `-w {e|w|n|p|v}` – specifies a validation level applied to the job to be submitted: e[error], w[arning], n[one], p[oke], v[erify]. Default: 'none'

# Requestable Resources (Complexes)

- <queue-type>.c: If not specified, normal.q queue is assumed.
- ram.c: If not specified, 5G per slot is assumed. Will be deprecated in favor of h_vmem.
- h_vmem – specifies maximum amount of memory all job's processes are allowed to use. This job will be killed if attempted to exceed.
- s_vmem – same as h_vmem, but will send USR1 signal.
- h_rt – specifies execution time hard limit. Currently, default is 12 hours. If runtime is exceeded, the job is killed. The shorter the requested time, the more chances there will be for the job to be dispatched sooner by means of the *backfilling* mechanism.
- s_rt – specifies execution time soft limit. USR1 signal is sent. Can be trapped with a script to log necessary information.
- hostname (h) – specifies a compute node on which this job should run

# Useful Commands

- qsub – submit a job
- qalter – modify parameters of an already submitted job, which is not yet running
- qdel – delete a job
- qmod – modify a job: suspend, clear error, re-schedule, etc.
- qhold – put/remove hold on a job
- qlogin – submit an interactive login session (currently not available on crius)
- qhost – show information about execution nodes
- qstat – show various runtime information about the cluster, queues, jobs, etc.
- qconf – show cluster configuration

# Additional Information

- Manual pages (man):
  - qsub, qlogin, qalter
  - qdel
  - qhost
  - qmod
  - qconf
  - qhold
  - qstat
  - sge_intro
  - sge_pe
  - complex

- Documentation
  - **http://docs.jgi-psf.org/UGE**
  - Coming soon: http://www.nersc.gov/users/computational_systems/phoebe-crius

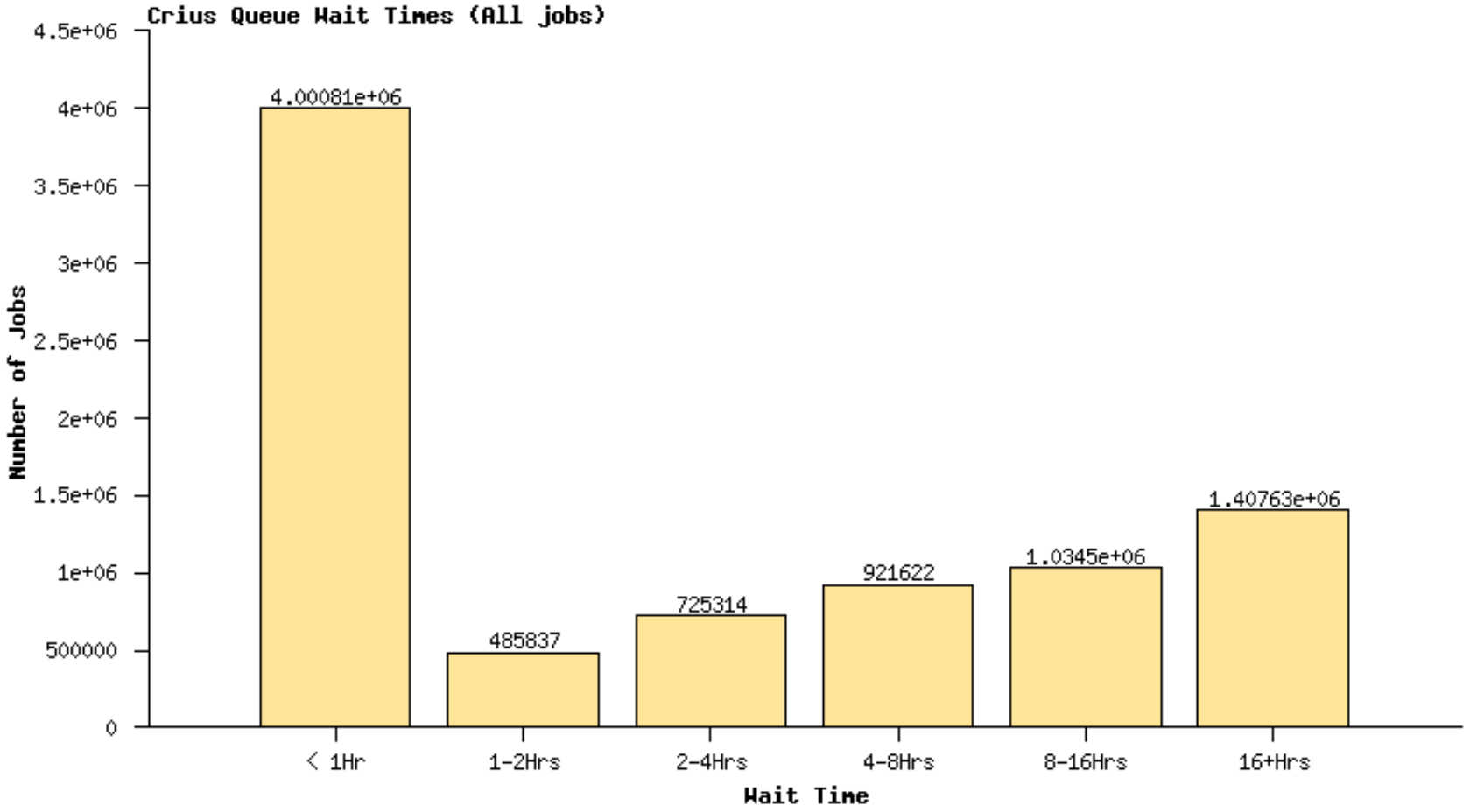
# Mailing List

All information regarding the clusters, *i.e.*, updates, changes, maintenance, etc., is distributed via the mailing list:

**sge@lists.jgi-psf.org**

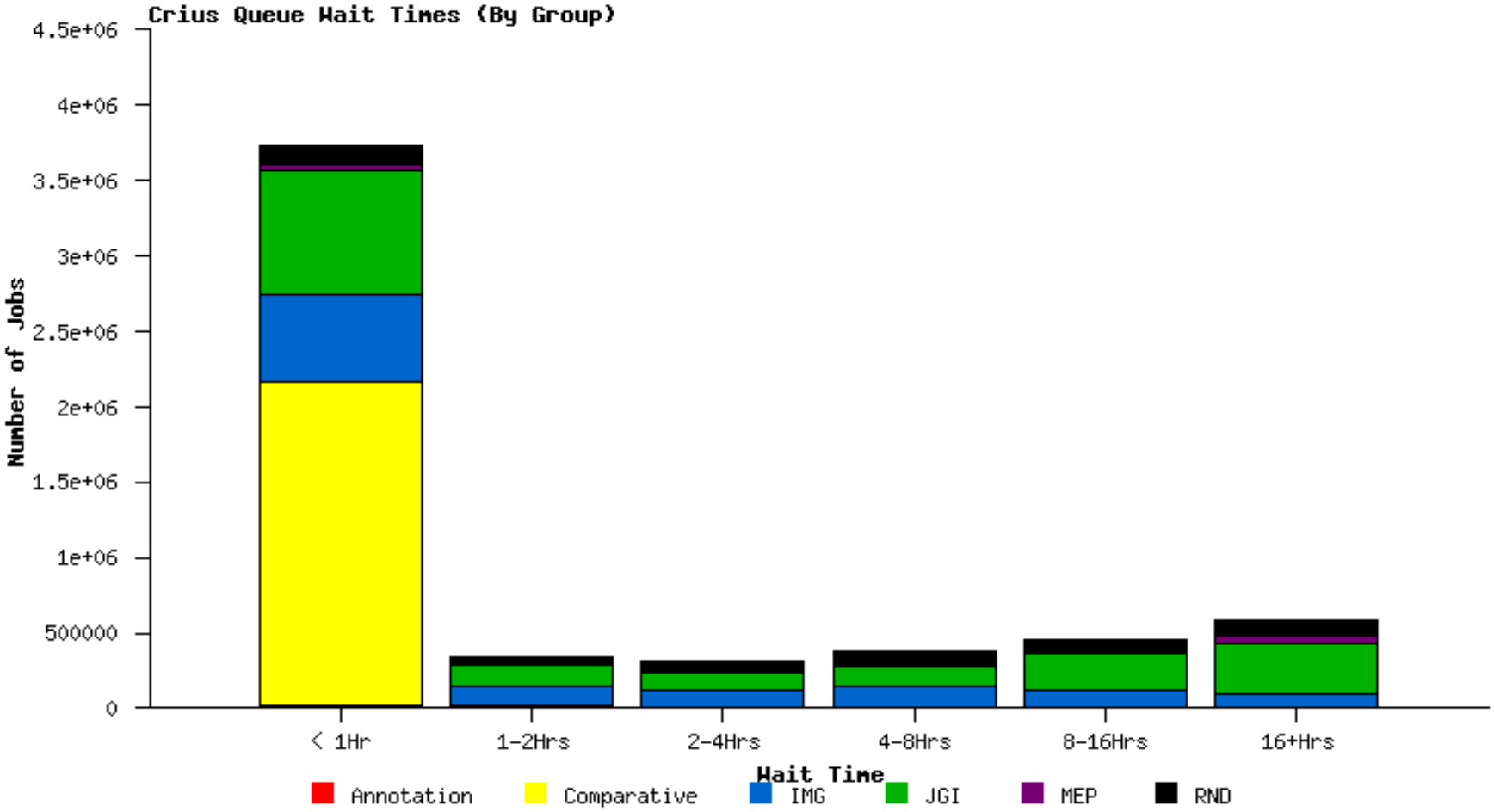
Self-subscription service is at **http://lists.jgi-psf.org**

# Queue Wait Times (All jobs)



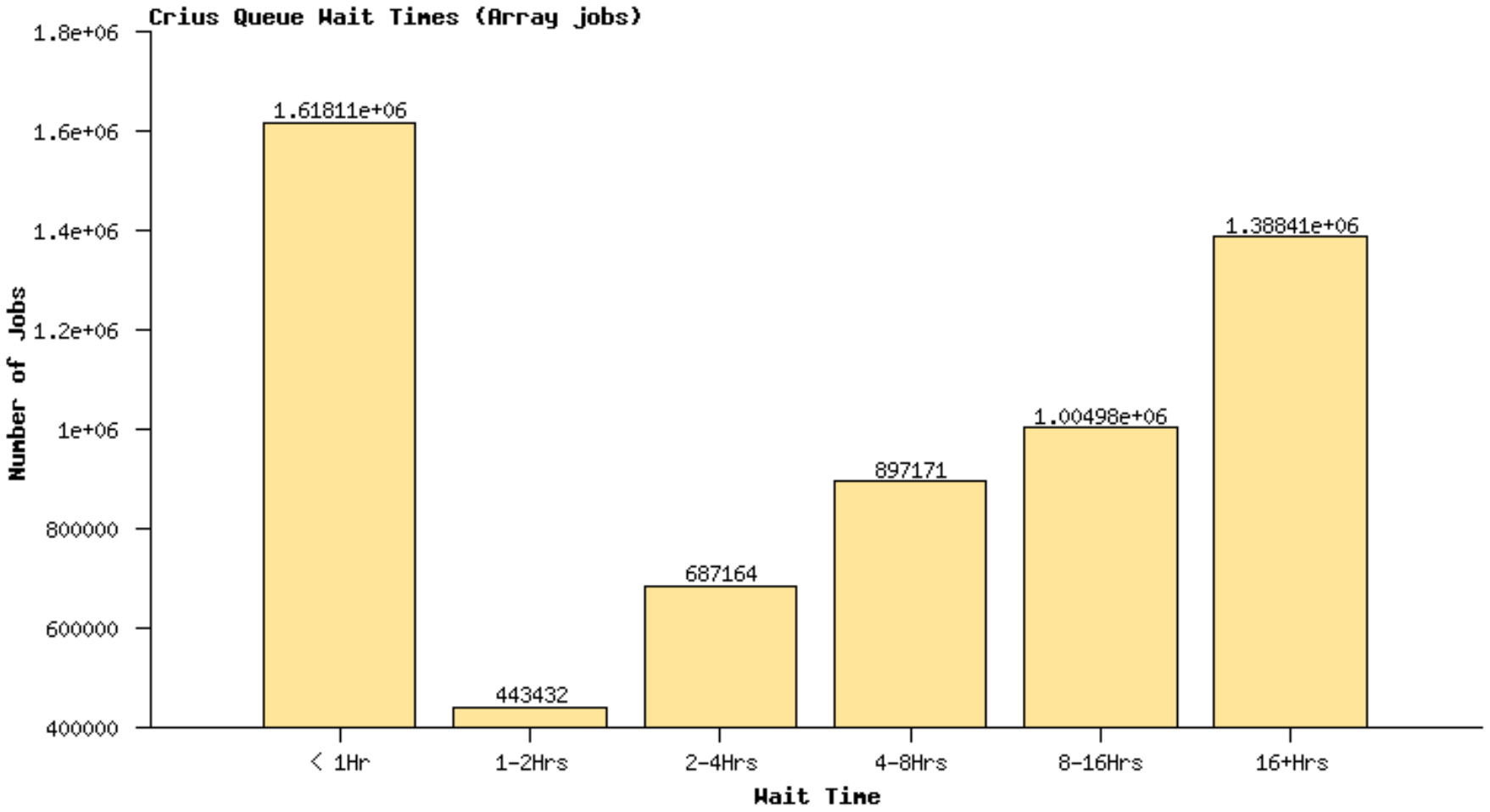

Late-Dec – Early Feb

# Queue Wait Times (By Project)



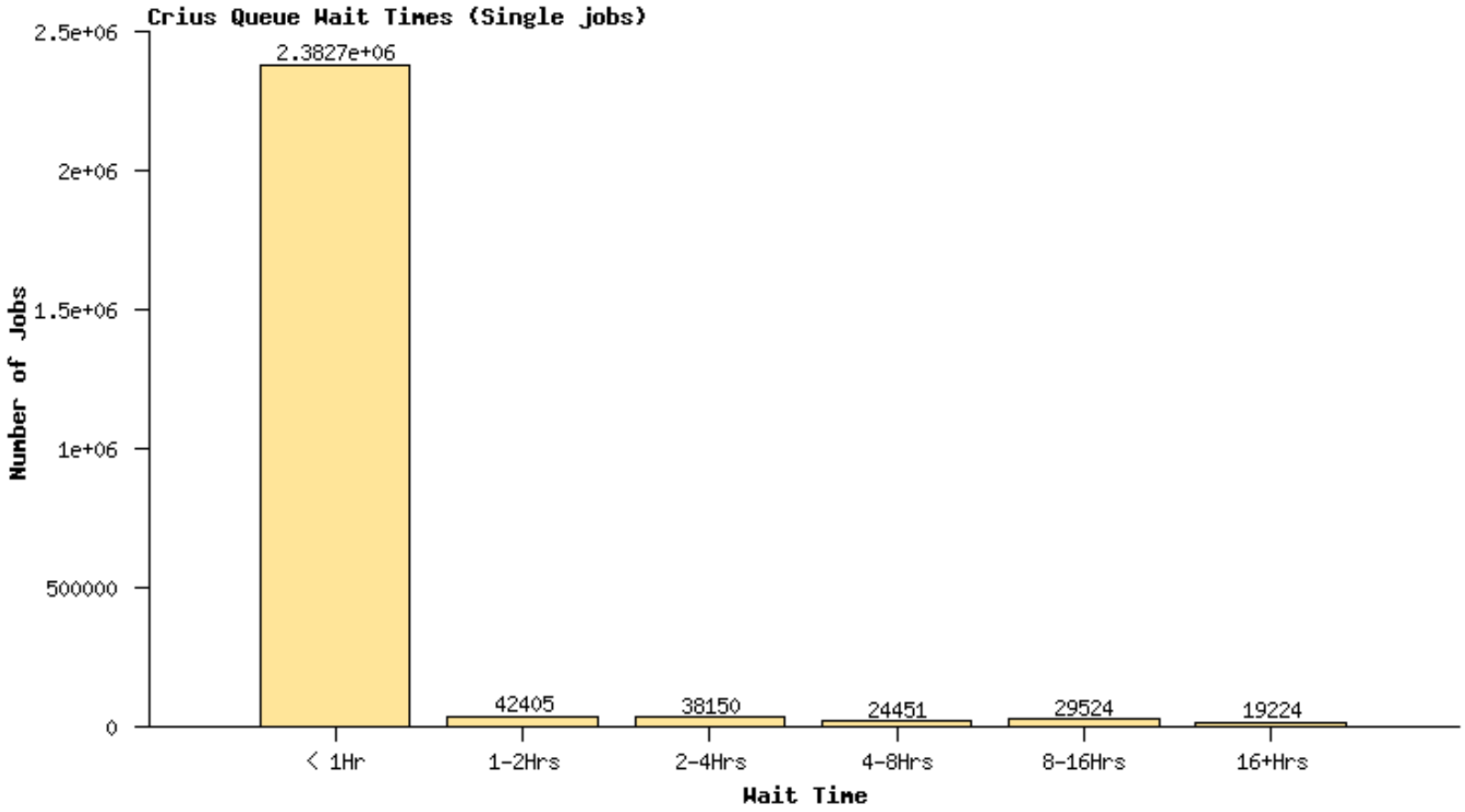

Late-Dec – Early Feb

# Queue Wait Times (Array jobs)




Late-Dec – Early Feb

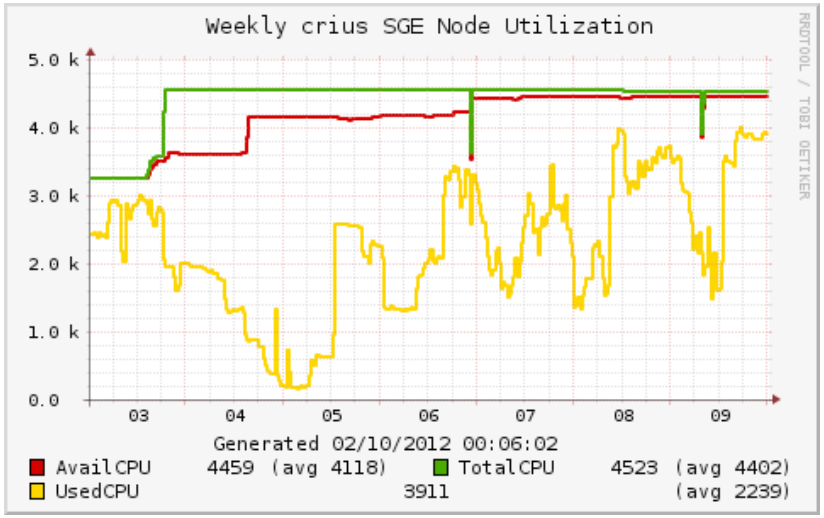
# Queue Wait Times (Single jobs)

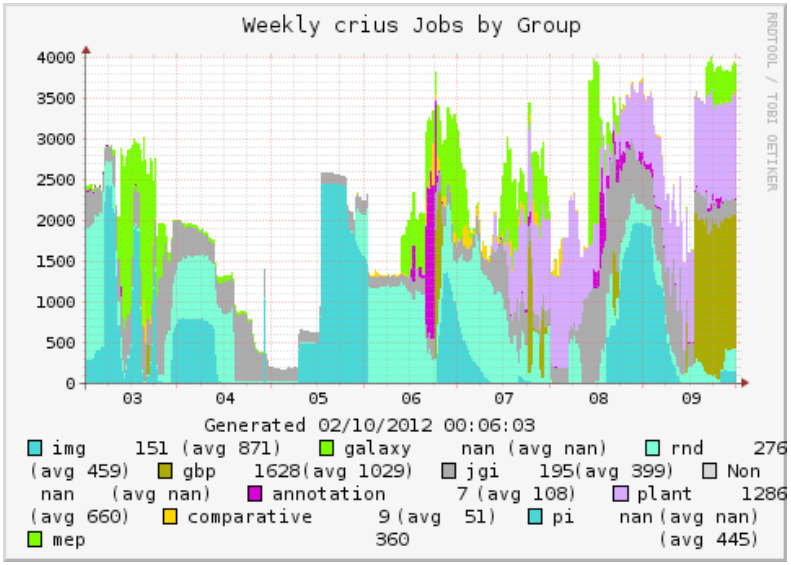



Late-Dec – Early Feb

# System and Share Utilization

- We collect metrics that track system utilization, share usage and queue wait-times
- We are finalizing the web display of these graphs, and they will be accessible to all users from the NERSC webpages



# Share Utilization