

Batch Jobs

login Environment

To be able to use the commands to submit batch jobs, add the following 2 lines to your .login file:

```
source /afs/slac/g/suncat/gpaw/setupenv
setenv PATH ${PATH}:/afs/slac/g/suncat/bin:/usr/local/bin
```

The first line sets up a default interactive "gpaw-friendly" environment (killing any earlier environment settings!). You could use a similar line to pick up a default "jacapo-friendly" environment, if you prefer. The second line adds some necessary interactive commands (e.g. for submitting batch jobs).

If you want to use a particular version (e.g. 27) of GPAW instead of the "default" above, use something like this instead:

```
source /nfs/slac/g/suncatfs/sw/gpawv27/setupenv
```

Note that the contents of .login/.cshrc do NOT affect batch job environment submitted with the various job submission commands described below (e.g. gpaw-bsub, jacapo-bsub, etc.).

Queues

Queue Name	Comment	Wallclock Duration
suncat-test	16 cores, for quick "does-it-crash" test	10 minutes
suncat-short		2 hours
suncat-medium		20 hours
suncat-long		50 hours
suncat-xlong	Requires Thomas/JensN/Frank/Felix permission. May have to limit time with -W flag	20 days

There are similar queue names for the suncat2/suncat3 farms.

Farm Information

Farm Name	Cores (or GPUs)	Cores (or GPUs) Per Node	Memory Per Core (or GPU)	Interconnect	Cost Factor	Notes
suncat	2272 Nehalem X5550	8	3GB	1Gbit Ethernet	1.0	
suncat2	768 Westmere X5650	12	4GB	2Gbit Ethernet	1.1	
suncat3	512 Sandy Bridge E5-2670	16	4GB	40Gbit QDR Infiniband	1.8	
suncat4	1024 Sandy Bridge E5-2680	16	2GB	1Gbit Ethernet	1.5	
gpu	119 Nvidia M2090	7	6GB	40Gbit QDR Infiniband	N/A	

Jobs should typically request a multiple of the number of cores per node.

Submitting Jobs

It is important to have an "afs token" before submitting jobs. Check the status with the **tokens** commands. Renew every 24 hours with **/usr/local/bin/kinit** command.

Login to a suncat login server (suncatls1,suncatls2 all @slac.stanford.edu) to execute commands like these (notice they are similar for gpaw/dacapo/jacapo):

```
gpaw-bsub -o mo2n.log -q suncat-long -n 8 mo2n.py
dacapo-bsub -o Al-fcc-single.log -q suncat-long -n 8 Al-fcc-single.py
jacapo-bsub -o Al-fcc-single.log -q suncat-long -n 8 co.py
```

You can find more

You can select a particular version to run (documented on the appropriate calculators page):

```
gpaw-ver-bsub 19 -o mo2n.log -q suncat-long -n 8 mo2n.py
```

You can also embed the job submission flags in your .py file with line(s) like:

```
#LSF -o mo2n.log -q suncat-long  
#LSF -n 8
```

The job submission scripts use the flags from both the command line and the .py file ("logical or").

Batch Job Output

Because of a file-locking bug in afs, **all output from our MPI jobs (GPAW, dacapo, jacapo) should go to nfs**. Our fileservers space is at **/nfs/slac/g/suncatfs**. Make a directory there with your username. You should always use the "/nfs" form of that name (the nfs automounter software often refers to it as "/a", but that syntax should not be in any of your scripts).

Batch Job Environment

The above commands "take control" and set all the environment, preventing the user from changing part of the environment (PATH, PYTHONPATH, etc.). If you want to take that fancier (but more error prone) approach, look at the 2 lines in the gpaw-bsub/dacapo-bsub scripts in /afs/slac/g/suncat/bin, and modify the environment after executing the "setupenv" command, and before executing the "bsub" command.

Useful Commands

Login to a suncat login server (suncatls1,suncatls2) to execute these. You can get more information about these commands from unix man pages.

```
bjobs (shows your current list of batch jobs and jobIds)  
bjobs -d (shows list of your recently completed batch jobs)  
bqueues suncat-long (shows number of cores pending and running)  
bjobs -u all | grep suncat (show jobs of all users in the suncat queues)  
bpeek <jobId> (examine logfile output from job that may not have been flushed to disk)  
bkill <jobId> (kill job)  
btop <jobId> (moves job priority to the top)  
bbot <jobId> (moves job priority to the bottom)  
bsub -w "ended\{12345\}" (wait for job id 12345 to be EXITed or DONE before running)  
bmod [options] <jobId> (modify job parameters after submission, e.g. priority (using -sp flag))  
bswitch suncat-xlong 12345 (move running job id 12345 to the suncat-xlong queue)  
bmod -n 12 12345 (change number of cores or pending job 12345 to 12)  
bqueues -r suncat-long (shows each user's current priority, number of running cores, CPU time used)  
bqueues | grep suncat (allows you to see how many pending jobs each queue has)
```

suncat4 Guidelines

These experimental computing nodes have relatively little memory. Please use the following guidelines when submitting jobs:

- if you exceed the 2GB/core memory limit, the node will crash. planewave codes (espresso, dacapo/jacapo, vasp) use less memory. If you use GPAW make sure you check the [memory estimate](#) before submitting your job. Here's some experience from Charlie Tsai on what espresso jobs can fit into a node:

```
For the systems I'm working with approximately 2x4x4 (a support that's 2x4x3, catalyst  
is one more layer on top) is about as big a system as I can get without running out of  
memory. For spin-polarized calculations, the largest system I was able to do was about  
2x4x3 (one 2x4x1 support and two layers of catalysts).
```

- you can observe the memory usage of the nodes for your job with "lsload psanacs002" (if your job uses node "psanacs002"). The last column shows the free memory.
- use the same job submission commands that you would use for suncat/suncat2
- use queue name "suncat4-long"
- the "-N" batch option (to receive email on job completion) does not work