

Batch Compute Best Practices and Other Info

▪ Specify an output file

Use the `-o` or `-oo` option to `bsub` to specify an output file for your batch job. If you do not specify a viable file, the output will be sent via email which when multiplied by the 100s or 1000s can easily overwhelm the mail server.

▪ Use local scratch space

Use local scratch space during the course of your job and copy over the results to the final location when the job finishes. Using local `/scratch` space is more efficient for constant writing and/or reading than `nfs` or `AFS` (i.e. over the network), and multiple batch jobs reading/writing to one `nfs` or `AFS` server can cause problems for the fileserver and everyone else who may be using it.

The amount of scratch available varies by machine type, but you can request a host with a certain amount of space with the `-R` option, e.g. `bsub -R "scratch>10"` would request a host with 10GB of scratch. A wrapper to submit your batch job is recommended which does the following:

- Create a directory in `/scratch` using the batch job ID (`$LSB_JOBID`).
- Copy any required input files to your `/scratch` directory.
- Write your program output to the newly created directory.
- When the program/script/command finishes copy the output file to a more permanent location.
- Remove your job directory from the scratch area.

▪ Specify a wall clock time

Use the `-W` or `-We` option to `bsub` to specify the length of time you expect your job to run. This is preferable to specifying a queue because it allows the batch system to schedule your job more efficiently, taking advantage of backfill opportunities.

▪ Getting the amount of memory needed for your job

There is a memory limit on the general queues of 4GB ram per-core and 10GB swap per-core. If your job needs more, you should request more cores with something like:

```
bsub -W <runlimit> -n 2 -R "span[hosts=1]" <your job>
```

This will submit a job to the general queues, using 2 cores on a single host which will give you a total memory limit of 8GB.

For queues that are not part of the general farm there is no such limit. You can use the `-R` and `-M` options to `bsub` to specify the amount of memory that you expect your job to use. For example,

```
bsub -q <non general farm queue> -R "mem > 15000" -M 15000 <yourjob>
```

The `-R` option tells `bsub` to schedule the job on a system with at least 15GB of memory available. The `-M` option tells LSF to allow your job to use up to 15GB of memory; if your job exceeds that it will be killed. Without this option, if your job exceeds the amount of memory available to the server, it will crash and take down all jobs currently running. Since your job may be running along with jobs from other users, you would be hurting not only yourself, but others as well.

Other Info:

- To access the anaconda version of python in your batch jobs, along with the correct libraries, set your `PATH` environment variable in your `.bashrc`:

```
export PATH=/afs/slac.stanford.edu/package/anaconda/anaconda/bin:$PATH
```