BSUB CHEATSHEET

Login and submitting jobs

To login first you should either connect to ETHZ's network or VPN. Then connect by ssh to euler.ethz.ch, using your nethz account and password.

To submit your jobs, first you should load the needed module, and use following command

bsub command [arguments]

While you could run light mission on ssh interface, It is require to submit heavy mission by the bsub system.

Example

zip a file

```
[leonhard@euler03 ~]$ bsub gzip big_file.dat
Generic job.
Job <8146539> is submitted to queue
    <normal.4h>.
```

Run a R file

```
bsub [LSF options] "R --vanilla --slave <
    inputfile.R > outputfile"
```

Runing a bash

```
[leonhard@euler03 ~]$ bsub < hello.sh
Generic job.
Job <8146544> is submitted to queue
    <normal.4h>.
```

Batch interactive job

```
[leonhard@euler03 ~]$ bsub -I "env | sort"
Generic job.
Job <8146545> is submitted to queue
    <normal.4h>.
<<Waiting for dispatch ...>
```

Resource requirement, using the parameter

```
[leonhard@euler03 ~]$ bsub -W 20:00
    ./solve_Koenigsberg_bridge_problem
Generic job.
Job <8146547> is submitted to queue
    <normal.24h>.
```

Submitting parameter I

You could always turn to https://scicomp.ethz.ch/lsf_submission_line_advisor for advice.

-W HH:MM	Wall-clock time required by the job. Can also be expressed in minutes.
-n N	Number of processors required by the job.
-R ”rusage[mem=X]”	Amount of memory (in MB per processor) required by the job.
-o outfile	Append the job's output (stdout) to outfile. The keyword ”% J”is interpreted as the job's numerical ID.
-e errfile	Append the job's error (stderr) to errfile. By default, stderr is merged with stdout.
-oo outfile	Write the job's output (stdout) to outfile, overwriting it if it already exists.
-eo errfile	Write the job's error (stderr) to errfile, overwriting it if it already exists.
-I / -Ip / -Is	Run the job interactively. Input/output are redirected from/to your terminal. Use -Ip to create a pseudo-terminal, and -Is to enable shell support.
-J jobname	Assign a (non necessarily unique) name to the job. Used to define job chains. To avoid confusion with numerical job IDs, jobname should contain at least one letter.
-w ”depcond”	Wait (do not start the job) until the specified dependency condition is satisfied. For example: ”done(jobID)”, ”ended(jobname)”. Quotes are recommended.
-B / -N	Send an e-mail to the job's owner (username@ethz.ch) when the job begins / ends.

Submitting parameter II

-u user	Send e-mail to user instead of the job's owner. The recipient's address must be inside the ETH domain. The firewall blocks e-mail sent to other addresses. (Note: This switch alone does not imply -B nor -N.)
-r	Indicate that the job is re-runnable. If the compute node where your job is running crashes, LSF will automatically re-run it from the beginning on a different node.
-G share_name	Use the share_name shareholder share to run this job

Job monitoring

busers	user limits, number of pending and running jobs
bqueues	queues status (open/closed; active/inactive)
bjobs	more or less detailed information about pending, running and recently finished jobs
bbjobs	better bjobs (bjobs with human readable output) some parameter: -r running, -d ended recently, -p show only pending jobs, -a all jobs
bhist	information about jobs that finished in the last hours/days
bpeek	display the standard output of a given job
lsf_load	show the CPU load of all nodes used by a job
bjob_connect	login to a node where one of your jobs is running
bkill	kill a job

Loading Module

You could load the installed module provided by the Euler System
The provided module could be found from https://scicomp.ethz.ch/wiki/Euler_applications

module list	Show currently loaded module
module load gcc/4.8.2	loaded gcc with given version
module purge	unload all loaded module