

Tip: Go to View > Show Document Outline to see a list of headings in this document.

## SLAC

1. Obtain a SLAC account
2. Request access to bulletmpi and bulletmpi-large queues
  - a. Send an email to [unix-admin@slac.stanford.edu](mailto:unix-admin@slac.stanford.edu)
3. Log in
  - a. `ssh username@rhel6-64.slac.stanford.edu`
  - b. `ssh bullet0001/bullet0002` (the head node doesn't have a right mpi module)
4. Environment setup
  - a. `. scl_source enable devtoolset-3 # includes GCC 4.9`
5. Use LSF to submit jobs
  - a. `bsub -q QUEUE -W HH:MM -n CORES -o OUTPUT_FILE command [args]`
  - b. FIXME: I have not found how to submit jobs via a script
6. Documentation
  - a. <https://confluence.slac.stanford.edu/display/SCSPub/High+Performance+Computing+at+SLAC>

## Summit (ORNL)

Summary: Power9, Volta and IBV with LSF (bsub/jsrun)

1. Obtain an ORNL account
  - a. Contact your sponsor for instructions
  - b. You will be mailed an OTP access token, the token must be taken to a notary public in order to be confirmed and activated
  - c. You will be asked to set a PIN
2. Log in
  - a. `ssh username@summit.olcf.ornl.gov`
  - b. Passcode is your PIN followed by an OTP code
3. Environment setup
  - a. `module load cmake/3.9.2`  
`module load gcc/6.4.0`  
`module load cuda/9.1.85`
4. Running jobs
  - a. Use LSF (bsub) to submit batch scripts
  - b. Use jsrun within a batch script to run jobs
5. GASNet bug 3447
  - a. **Note: This should be fixed in GASNet 2019.6.0 or newer**
  - b. There is a GASNet bug associated with the use of multiple HCAs per node
  - c. To avoid this, please run with the script attached to the bug report  
[https://upc-bugs.lbl.gov/bugzilla/show\\_bug.cgi?id=3447](https://upc-bugs.lbl.gov/bugzilla/show_bug.cgi?id=3447)
6. Documentation

- a. <https://www.olcf.ornl.gov/for-users/system-user-guides/summit/>

## Cori (NERSC)

Summary: Cray (Aries) with Native SLURM

7. Obtain a NERSC account
  - a. [https://nim.nersc.gov/nersc\\_account\\_request.php](https://nim.nersc.gov/nersc_account_request.php)
  - b. After getting an account, contact your PI to be given access to the appropriate project(s)
8. Log in
  - a. `ssh username@cori.nersc.gov`
9. Environment setup
  - a. `module unload PrgEnv-intel`  
`module load PrgEnv-gnu`  
`export CC=cc`  
`export CXX=CC`

## Surface & Quartz (LLNL)

10. Obtain an LLNL account
  - a. Apply via Sarape  
([https://rails-rn-prod.sandia.gov/sarape/help/request\\_submitted/](https://rails-rn-prod.sandia.gov/sarape/help/request_submitted/))
  - b. You must be on the Stanford network to apply
  - c. You will be mailed an OTP access token
  - d. You will be asked to set a PIN
  - e. You will receive two **different** usernames: an OUN and a LC username
11. Log in
  - a. Use one of:  
`ssh username@surface.llnl.gov`  
`ssh username@quartz.llnl.gov`
  - b. Username is your **LC username**, NOT OUN
  - c. Password is your PIN followed by an OTP code
12. Environment setup
  - a. Surface:  
`use git-2.0.0`  
`use gcc-4.9.3p`  
`use mvapich2-gnu-2.2`
  - b. Quartz:  
`module load gcc/4.9.3`  
`module load openmpi/2.0.0`  
*# git-2.8 is configured by default*
13. Use Moab to submit jobs and srun or mpirun to execute programs
  - a. <https://computing.llnl.gov/tutorials/moab/#JobScript>

- b. Note: PSM (OmniPath) only supports one endpoint per node, so if you use GASNet with MPI compatibility (which we usually do) you need to make sure OpenMPI is configured to NOT use the PSM conduit itself
  - c. Quartz:
 

```
export OMPI_MCA_mtl="^psm,psm2"
srun -n NODES --tasks-per-node 1 --cpu_bind none --mpibind=off CMD
```
  - d. Note: Quartz uses the mpibind plugin for SLURM to handle core assignment. This means that many of the usual SLURM flags don't work. If you just want one runtime per node, you can turn it off as above. Otherwise you need to set OMP\_NUM\_THREADS to 2x the number of cores you want (so that you get hyperthreads) and remove the other flags. E.g.:
 

```
OMP_NUM_THREADS=36 srun -N <NODES> -n <2*NODES> <CMD>
```

 (Gives each runtime 18 physical cores, i.e. 2 hyperthreads per core, so that you get one runtime per socket.)
14. Documentation
- a. <https://computing.llnl.gov/resources>

## Lassen & Sierra (LLNL)

15. Obtain an LLNL account
- a. Apply via Sarape  
([https://rails-rn-prod.sandia.gov/sarape/help/request\\_submitted/](https://rails-rn-prod.sandia.gov/sarape/help/request_submitted/))
  - b. You must be on the Stanford network to apply
  - c. You will be mailed an OTP access token
  - d. You will be asked to set a PIN
  - e. You will receive two **different** usernames: an OUN and a LC username
16. Log in
- a. Requires two steps:
 

```
ssh username@oslic.llnl.gov
ssh lassen
```
  - b. Username is your **LC username**, NOT OUN
  - c. Password is your PIN followed by an OTP code
17. Environment setup
- a. 

```
module load gcc/7.3.1
module load cmake
```
18. Regent setup
- a. 

```
CONDUIT=ibv ./scripts/setup_env.py --llvm-version 60 --terra-url
https://github.com/StanfordLegion/terra.git --terra-branch luajit2.1-ppc64
```
19. Use bsub to submit jobs, jsrun to launch commands
- a. The official documentation recommends Irun, but jsrun is more portable (e.g. it's the same command as on Summit)
20. Documentation
- a. <https://lc.llnl.gov/confluence/display/SIERRA/Sierra+Systems>

- b. <https://hpc.llnl.gov/training/tutorials/using-lcs-sierra-system>

## Piz Daint

Summary: Cray (Aries) with Native SLURM

1. Obtain a Piz Daint account
  - a. <https://www.cscs.ch/user-lab/applying-for-accounts/>
  - b. Alex Aiken is the PI
2. Log in
  - a. Two step process:  
ssh username@ela.cscs.ch  
ssh username@daint.cscs.ch
  - b. Note: The machine accepts SSH keys
3. Environment setup
  - a. module load daint-gpu  
module load git/2.11.0  
module unload PrgEnv-cray  
module load PrgEnv-gnu/6.0.3  
export CC=cc  
export CXX=CC
4. Use SLURM to submit jobs
5. Documentation
  - a. <https://user.cscs.ch/>

## Sapling

1. See <https://github.com/StanfordLegion/sapling-guide>

## XStream

2. Obtain an XStream account
  - a. Need a SUNetID to get access
3. Log in
  - a. First time only: Add the following to ~/.ssh/config  
Host xstream-login.stanford.edu  
GSSAPIAuthentication yes  
GSSAPIDelegateCredentials yes  
(from <https://xstream.stanford.edu/docs/start/>)
  - b. kinit username@stanford.edu
  - c. ssh username@xstream-login.stanford.edu
4. Environment setup
  - a. module load GCC/4.9.2-binutils-2.25
  - b. module load CUDA/7.5.18

- c. `module load cuDNN/5.1-CUDA-7.5.18`
- d. `module load OpenMPI/1.8.5`
- e. `export CONDUIT=ibv`  
`git clone https://github.com/StanfordLegion/gasnet.git`  
`cd gasnet`  
`make`  
`export GASNET=$PWD/release`
- 5. Use SLURM to submit jobs
  - a. Note: `srun` is configured NOT to pass environment variables by default. Use the following flags to pass the necessary Regent/CUDA variables  
`srun --export=TERRA_PATH,INCLUDE_PATH,LD_LIBRARY_PATH`  
`srun --export=CUDA_HOME`
- 6. Documentation
  - a. <http://xstream.stanford.edu>