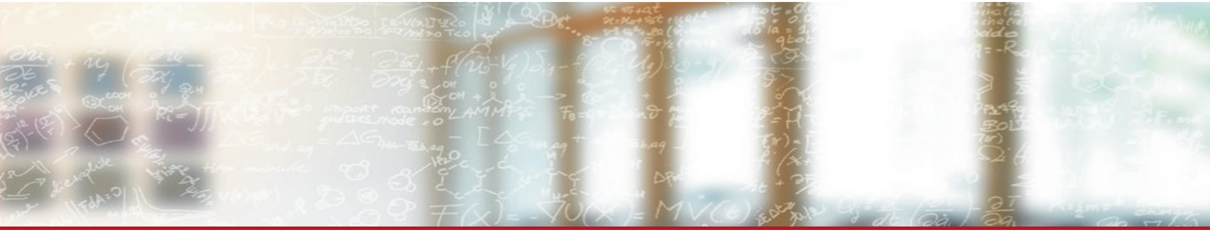
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Introduction to the Piz Daint environment

Directive Based GPU Programming

Vasileios Karakasis, CSCS

May 14–15, 2018

Overview

- Accessing CSCS
- Compiling my code
- Running my code
- Editing my code
- Transferring files from/to CSCS
- Repository of the course

Piz Daint

Computing nodes

Piz Daint is a hybrid cluster of Cray XC40/XC50 nodes

- Hybrid nodes (XC50)
  - 5320 total
  - Intel® Xeon® E5-2690 v3 @ 2.60GHz (12 cores, 64GB RAM, Haswell)
  - NVIDIA® Tesla® P100 16GB (Pascal)
- Multicore nodes
  - 1813 nodes
  - Two Intel® Xeon®E5-2695 v4 @ 2.10GHz (2 x 18 cores, 64/128 GB RAM, Broadwell)
- Login nodes
  - 5 total
  - Intel® Xeon®CPU E5-2650 v3 @ 2.30GHz (10 cores, 256 GB RAM, Haswell)
- Aries routing and communications ASIC, and Dragonfly network topology

Piz Daint

Filesystems

- `/scratch`: High performance Lustre filesystem accessible from the computing nodes
  - Environment variable `$SCRATCH` points to it
  - Total capacity: 6.2 PB
  - Must be used for heavy I/O
- `/users`: GPFS filesystem for the users' homes
- `/project`, `/store`: Long-term storage for computational projects

More on `https://user.cscs.ch/storage/file_systems/`

Accessing Piz Daint

- Accessible through SSH
- Piz Daint is not directly accessible from the outside world:
  - `ela` → `daint10x` → `nidxxxxx`

Two-steps process:

1. Login to the frontend, forwarding X11 (will be needed the second day)
2. Move to the login nodes of Piz Daint

```
# Login to the frontend first
ssh -Y courseXX@ela.cscs.ch
ssh -Y daint
```

Programming Environments

Cray Linux Programming Environment

- 4 compilers available: `CCE`, `GNU`, `INTEL`, `PGI`
- 4 predefined Programming Environments:
  - PrgEnv-`cray` (default), `PrgEnv-gnu`, `PrgEnv-intel`, `PrgEnv-pgi`
  - `echo $PE_ENV` to get the current PrgEnv
- 3 wrappers available: `ftn` (Fortran), `cc` (C), `CC` (C++)
  - Required for compiling MPI programs
  - They set appropriate optimisation flags for the target architecture (CPU or GPU)
  - They provide a sort of portability across the programming environments

Managing programming environments

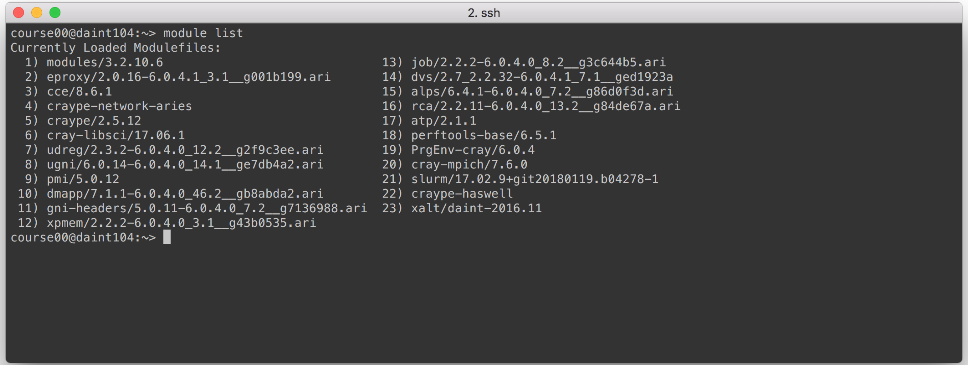
Daint uses *Environment Modules* (TMod) for managing the programming environments and the software packages:

- Dynamic modification of a user's environment via *modulefiles*.
- All programming environments and software on Daint is available through modules.
- The compiler wrappers will detect the loaded programming environment and automatically set the correct flags and libraries.

Managing programming environments

Listing modules

– `module list`

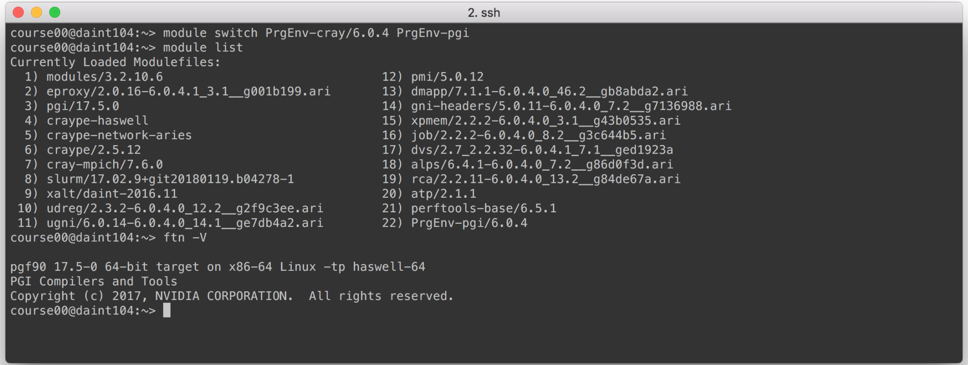


```
course00@daint104:~> module list
Currently Loaded Modulefiles:
  1) modules/3.2.10.6                           13) job/2.2.2-6.0.4.0_8.2__g3c644b5.ari
  2) eproxy/2.0.16-6.0.4.1_3.1__g001b199.ari    14) dvs/2.7_2.2.32-6.0.4.1_7.1__ged1923a
  3) cce/8.6.1                                  15) alps/6.4.1-6.0.4.0_7.2__g86d0f3d.ari
  4) craype-network-aries                       16) rca/2.2.11-6.0.4.0_13.2__g84de67a.ari
  5) craype/2.5.12                              17) atp/2.1.1
  6) cray-libsci/17.06.1                        18) perftools-base/6.5.1
  7) udreg/2.3.2-6.0.4.0_12.2__g2f9c3ee.ari     19) PrgEnv-cray/6.0.4
  8) ugni/6.0.14-6.0.4.0_14.1__ge7db4a2.ari     20) cray-mpich/7.6.0
  9) pmi/5.0.12                                 21) slurm/17.02.9+git20180119.b04278-1
 10) dmapp/7.1.1-6.0.4.0_46.2__gb8abda2.ari     22) craype-haswell
 11) gni-headers/5.0.11-6.0.4.0_7.2__g7136988.ari  23) xalt/daint-2016.11
 12) xpmem/2.2.2-6.0.4.0_3.1__g43b0535.ari
course00@daint104:~>
```

Managing programming environments

Switching programming environments

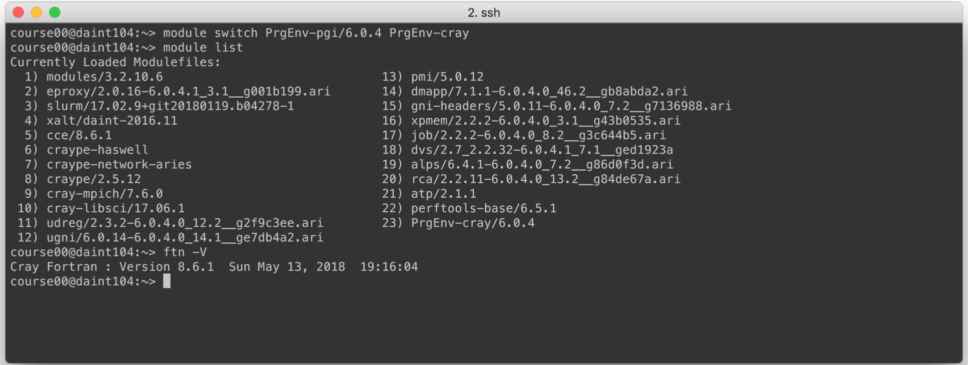
- Switch to the PGI programming environment
- `module switch`



```
course00@daint104:~> module switch PrgEnv-cray/6.0.4 PrgEnv-pgi
course00@daint104:~> module list
Currently Loaded Modulefiles:
  1) modules/3.2.10.6                          12) pmi/5.0.12
  2) eproxy/2.0.16-6.0.4.1_3.1__g001b199.ari   13) dmapp/7.1.1-6.0.4.0_46.2__gb8abda2.ari
  3) pgi/17.5.0                                14) gni-headers/5.0.11-6.0.4.0_7.2__g7136988.ari
  4) craype-haswell                            15) xpmem/2.2.2-6.0.4.0_3.1__g43b0535.ari
  5) craype-network-aries                      16) job/2.2.2-6.0.4.0_8.2__g3c644b5.ari
  6) craype/2.5.12                             17) dvs/2.7_2.2.32-6.0.4.1_7.1__ged1923a
  7) cray-mpich/7.6.0                          18) alps/6.4.1-6.0.4.0_7.2__g86d0f3d.ari
  8) slurm/17.02.9+git20180119.b04278-1        19) rca/2.2.11-6.0.4.0_13.2__g84de67a.ari
  9) xalt/daint-2016.11                        20) atp/2.1.1
 10) udreg/2.3.2-6.0.4.0_12.2__g2f9c3ee.ari    21) perftools-base/6.5.1
 11) ugni/6.0.14-6.0.4.0_14.1__ge7db4a2.ari    22) PrgEnv-pgi/6.0.4
course00@daint104:~> ftn -V

pgf90 17.5-0 64-bit target on x86-64 Linux -tp haswell-64
PGI Compilers and Tools
Copyright (c) 2017, NVIDIA CORPORATION.  All rights reserved.
course00@daint104:~> 
```

Managing programming environments

Switching back to the Cray programming environment



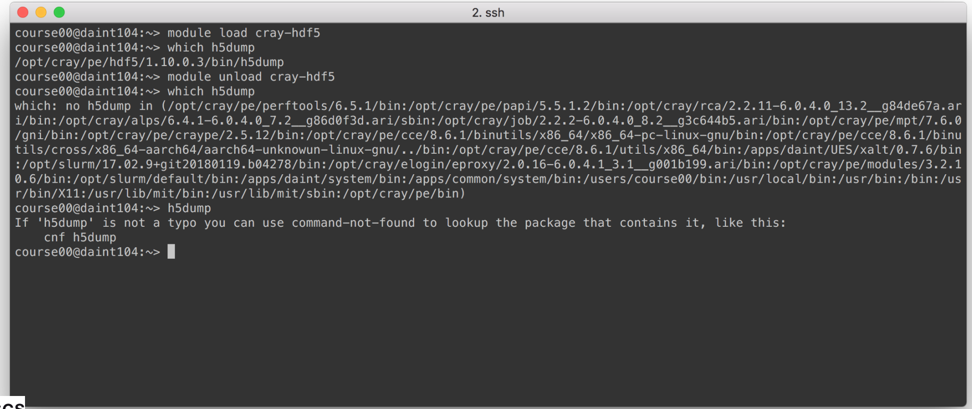
2. ssh

```
course00@daint104:~> module switch PrgEnv-pgi/6.0.4 PrgEnv-cray
course00@daint104:~> module list
Currently Loaded Modulefiles:
  1) modules/3.2.10.6                               13) pmi/5.0.12
  2) eproxy/2.0.16-6.0.4.1_3.1__g001b199.ari        14) dmapp/7.1.1-6.0.4.0_46.2__gb8abda2.ari
  3) slurm/17.02.9+git20180119.b04278-1             15) gni-headers/5.0.11-6.0.4.0_7.2__g7136988.ari
  4) xalt/daint-2016.11                             16) xpmem/2.2.2-6.0.4.0_3.1__g43b0535.ari
  5) cce/8.6.1                                      17) job/2.2.2-6.0.4.0_8.2__g3c644b5.ari
  6) craype-haswell                                 18) dvs/2.7_2.2.32-6.0.4.1_7.1__ged1923a
  7) craype-network-aries                           19) alps/6.4.1-6.0.4.0_7.2__g86d0f3d.ari
  8) craype/2.5.12                                  20) rca/2.2.11-6.0.4.0_13.2__g84de67a.ari
  9) cray-mpich/7.6.0                               21) atp/2.1.1
 10) cray-libsci/17.06.1                            22) perftools-base/6.5.1
 11) udreg/2.3.2-6.0.4.0_12.2__g2f9c3ee.ari         23) PrgEnv-cray/6.0.4
 12) ugni/6.0.14-6.0.4.0_14.1__ge7db4a2.ari
course00@daint104:~> ftn -V
Cray Fortran : Version 8.6.1  Sun May 13, 2018  19:16:04
course00@daint104:~> 
```

Managing programming environments

Loading and unloading modules

- `module load [MODULE_NAME]`
- `module unload [MODULE_NAME]`



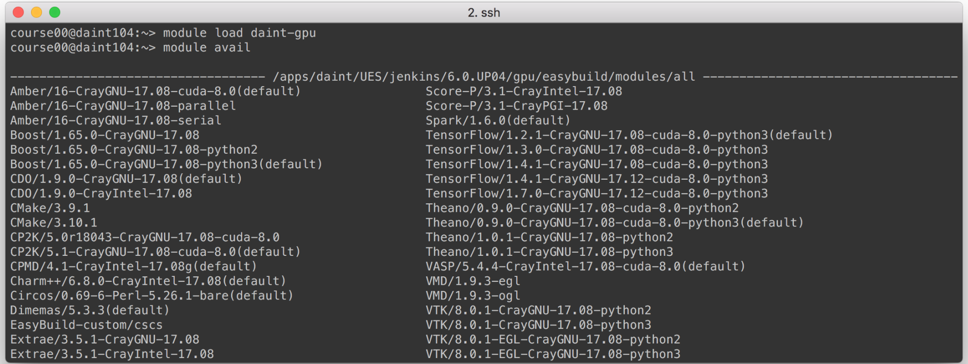
2. ssh

```
course00@daint104:~> module load cray-hdf5
course00@daint104:~> which h5dump
/opt/cray/pe/hdf5/1.10.0.3/bin/h5dump
course00@daint104:~> module unload cray-hdf5
course00@daint104:~> which h5dump
which: no h5dump in (/opt/cray/pe/perftools/6.5.1/bin:/opt/cray/pe/papi/5.5.1.2/bin:/opt/cray/rca/2.2.11-6.0.4.0_13.2__g84de67a.ar
i/bin:/opt/cray/alps/6.4.1-6.0.4.0_7.2__g86d0f3d.ari/sbin:/opt/cray/job/2.2.2-6.0.4.0_8.2__g3c644b5.ari/bin:/opt/cray/pe/mpt/7.6.0
/gni/bin:/opt/cray/pe/craype/2.5.12/bin:/opt/cray/pe/cce/8.6.1/binutils/x86_64/x86_64-pc-linux-gnu/bin:/opt/cray/pe/cce/8.6.1/binu
tils/cross/x86_64-aarch64/aarch64-unknowun-linux-gnu/../bin:/opt/cray/pe/cce/8.6.1/utils/x86_64/bin:/apps/daint/UES/xalt/0.7.6/bin
:/opt/slurm/17.02.9+git20180119.b04278/bin:/opt/cray/elogin/eproxy/2.0.16-6.0.4.1_3.1__g001b199.ari/bin:/opt/cray/pe/modules/3.2.1
0.6/bin:/opt/slurm/default/bin:/apps/daint/system/bin:/apps/common/system/bin:/users/course00/bin:/usr/local/bin:/usr/bin:/bin:/us
r/bin/X11:/usr/lib/mit/bin:/usr/lib/mit/sbin:/opt/cray/pe/bin)
course00@daint104:~> h5dump
If 'h5dump' is not a typo you can use command-not-found to lookup the package that contains it, like this:
    cnf h5dump
course00@daint104:~> 
```

Managing programming environments

Checking available modules

- The `daint-gpu` makes available the CSCS software stack built for the hybrid nodes of the system

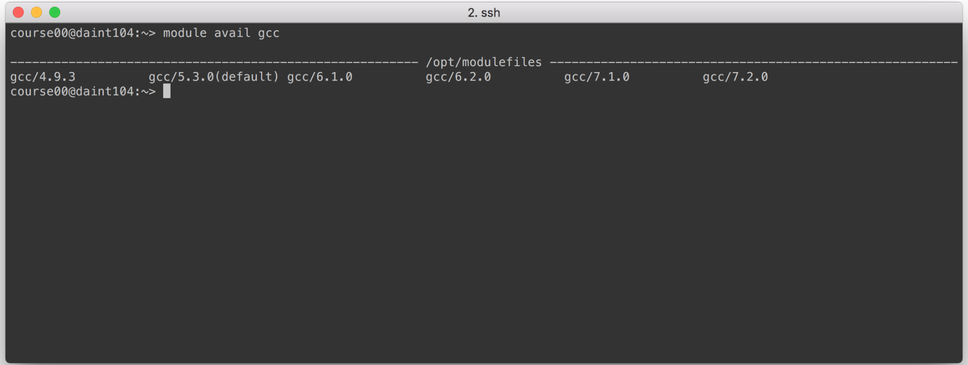


```
course00@daint104:~> module load daint-gpu
course00@daint104:~> module avail

----------------------------------- /apps/daint/UES/jenkins/6.0.UP04/gpu/easybuild/modules/all -----------------------------------
Amber/16-CrayGNU-17.08-cuda-8.0(default)                 Score-P/3.1-CrayIntel-17.08
Amber/16-CrayGNU-17.08-parallel                          Score-P/3.1-CrayPGI-17.08
Amber/16-CrayGNU-17.08-serial                            Spark/1.6.0(default)
Boost/1.65.0-CrayGNU-17.08                               TensorFlow/1.2.1-CrayGNU-17.08-cuda-8.0-python3(default)
Boost/1.65.0-CrayGNU-17.08-python2                       TensorFlow/1.3.0-CrayGNU-17.08-cuda-8.0-python3
Boost/1.65.0-CrayGNU-17.08-python3(default)              TensorFlow/1.4.1-CrayGNU-17.08-cuda-8.0-python3
CDO/1.9.0-CrayGNU-17.08(default)                         TensorFlow/1.4.1-CrayGNU-17.12-cuda-8.0-python3
CDO/1.9.0-CrayIntel-17.08                                TensorFlow/1.7.0-CrayGNU-17.12-cuda-8.0-python3
CMake/3.9.1                                              Theano/0.9.0-CrayGNU-17.08-cuda-8.0-python2
CMake/3.10.1                                             Theano/0.9.0-CrayGNU-17.08-cuda-8.0-python3(default)
CP2K/5.0r18043-CrayGNU-17.08-cuda-8.0                    Theano/1.0.1-CrayGNU-17.08-python2
CP2K/5.1-CrayGNU-17.08-cuda-8.0(default)                 Theano/1.0.1-CrayGNU-17.08-python3
CPMD/4.1-CrayIntel-17.08g(default)                       VASP/5.4.4-CrayIntel-17.08-cuda-8.0(default)
Charm++/6.8.0-CrayIntel-17.08(default)                   VMD/1.9.3-egl
Circos/0.69-6-Perl-5.26.1-bare(default)                  VMD/1.9.3-ogl
Dimemas/5.3.3(default)                                   VTK/8.0.1-CrayGNU-17.08-python2
EasyBuild-custom/cscs                                    VTK/8.0.1-CrayGNU-17.08-python3
Extrae/3.5.1-CrayGNU-17.08                               VTK/8.0.1-EGL-CrayGNU-17.08-python2
Extrae/3.5.1-CrayIntel-17.08                             VTK/8.0.1-EGL-CrayGNU-17.08-python3
```

Managing programming environments

Checking available modules

- Check available versions of a software

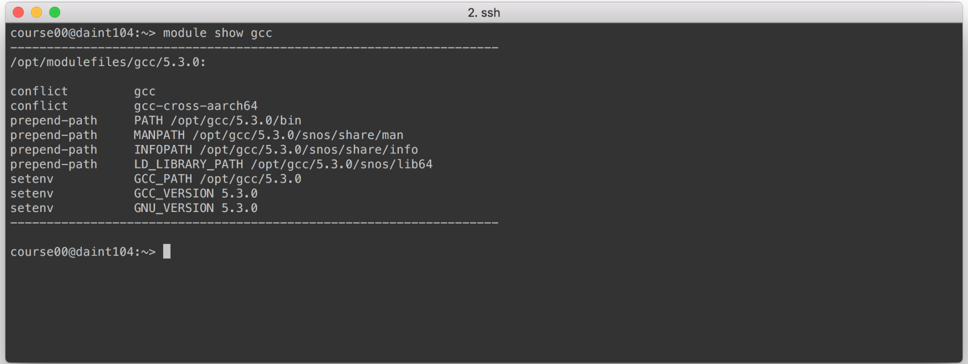


```
course00@daint104:~> module avail gcc

-------------------------------------------------------- /opt/modulefiles ---------------------------------------------------------
gcc/4.9.3          gcc/5.3.0(default) gcc/6.1.0          gcc/6.2.0          gcc/7.1.0          gcc/7.2.0
course00@daint104:~> 
```

Managing programming environments

Get information about a module

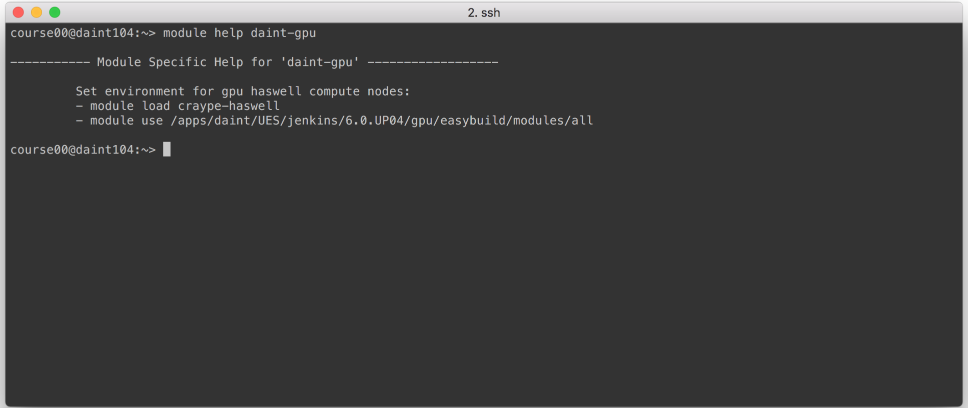
– Environment variables set, paths etc.



```
course00@daint104:~> module show gcc
-------------------------------------------------------------------
/opt/modulefiles/gcc/5.3.0:

conflict         gcc
conflict         gcc-cross-aarch64
prepend-path     PATH /opt/gcc/5.3.0/bin
prepend-path     MANPATH /opt/gcc/5.3.0/snos/share/man
prepend-path     INFOPATH /opt/gcc/5.3.0/snos/share/info
prepend-path     LD_LIBRARY_PATH /opt/gcc/5.3.0/snos/lib64
setenv           GCC_PATH /opt/gcc/5.3.0
setenv           GCC_VERSION 5.3.0
setenv           GNU_VERSION 5.3.0
-------------------------------------------------------------------

course00@daint104:~>
```

Managing programming environments

Get help for a module



```
course00@daint104:~> module help daint-gpu

----------- Module Specific Help for 'daint-gpu' ------------------

         Set environment for gpu haswell compute nodes:
         - module load craype-haswell
         - module use /apps/daint/UES/jenkins/6.0.UP04/gpu/easybuild/modules/all

course00@daint104:~>
```

Running on Piz Daint

The job scheduler

Piz Daint uses native SLURM for running jobs on the compute nodes. There are two ways of submitting a job:

1. Interactively from the login nodes using the `srun` command.
2. By submitting a job script using the `sbatch` command.

Running on Piz Daint

Using the `srun` command

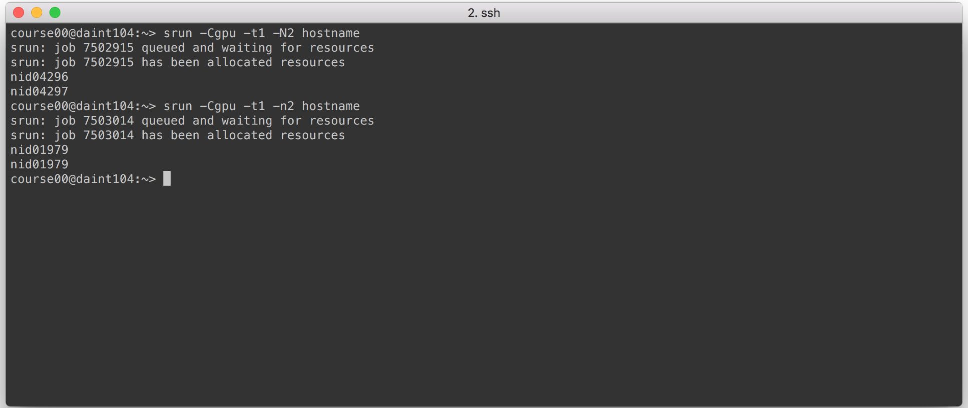
Necessary and useful options:

- `-C gpu`: requests allocation on the hybrid (GPU) nodes (required)
- `--reservation=openacc`: the reservation for our course to avoid waiting times
  - Reservation is valid for the two days of the course from 07:00-19:00.
- `-N 2`: number of compute nodes (default is 1)
- `-n 2`: number of MPI tasks (default is 1)
- `-t 5`: maximum duration of the job (default is 30min)
  - Allows to get an allocation quicker
  - Job will be killed if time limit is reached
  - Maximum time slot for a job is 24h

More on `https://user.cscs.ch/getting_started/running_jobs/`

Running on Piz Daint

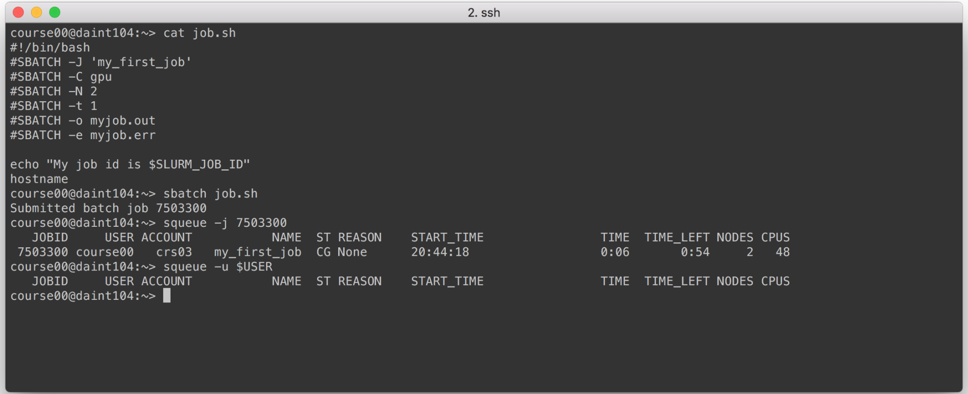
Using the `srun` command



```
course00@daint104:~> srun -Cgpu -t1 -N2 hostname
srun: job 7502915 queued and waiting for resources
srun: job 7502915 has been allocated resources
nid04296
nid04297
course00@daint104:~> srun -Cgpu -t1 -n2 hostname
srun: job 7503014 queued and waiting for resources
srun: job 7503014 has been allocated resources
nid01979
nid01979
course00@daint104:~>
```

Running on Piz Daint

Using the `sbatch` command



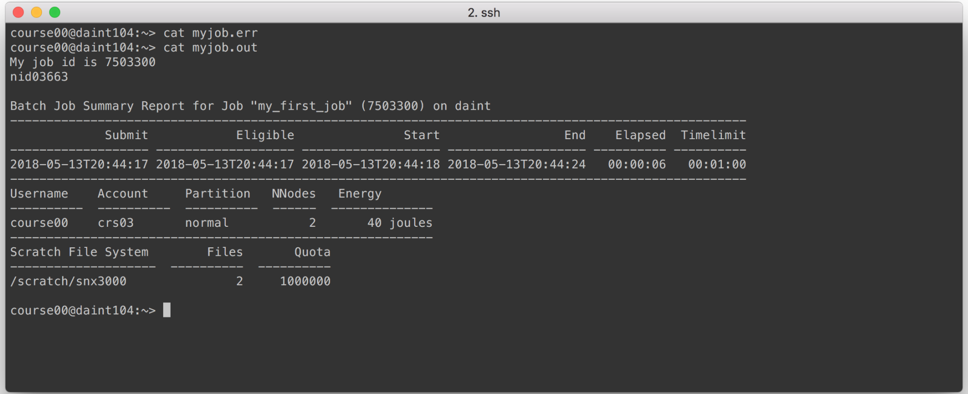
```
course00@daint104:~> cat job.sh
#!/bin/bash
#SBATCH -J 'my_first_job'
#SBATCH -C gpu
#SBATCH -N 2
#SBATCH -t 1
#SBATCH -o myjob.out
#SBATCH -e myjob.err

echo "My job id is $SLURM_JOB_ID"
hostname
course00@daint104:~> sbatch job.sh
Submitted batch job 7503300
course00@daint104:~> squeue -j 7503300
   JOBID     USER ACCOUNT           NAME  ST REASON    START_TIME                TIME  TIME_LEFT NODES CPUS
 7503300 course00   crs03   my_first_job  CG None      20:44:18                  0:06       0:54     2   48
course00@daint104:~> squeue -u $USER
   JOBID     USER ACCOUNT           NAME  ST REASON    START_TIME                TIME  TIME_LEFT NODES CPUS
course00@daint104:~> 
```

CSCS

ETH zürich

Running on Piz Daint

Using the `sbatch` command – Examinining the output



```
course00@daint104:~> cat myjob.err
course00@daint104:~> cat myjob.out
My job id is 7503300
nid03663

Batch Job Summary Report for Job "my_first_job" (7503300) on daint
-----------------------------------------------------------------------------------------------------
             Submit            Eligible               Start                 End    Elapsed  Timelimit
------------------- ------------------- ------------------- ------------------- ---------- ----------
2018-05-13T20:44:17 2018-05-13T20:44:17 2018-05-13T20:44:18 2018-05-13T20:44:24   00:00:06   00:01:00
-----------------------------------------------------------------------------------------------------
Username    Account     Partition   NNodes   Energy
----------  ----------  ----------  ------  --------------
course00    crs03       normal           2       40 joules
----------------------------------------------------------
Scratch File System        Files       Quota
-------------------- ---------- ----------
/scratch/snx3000              2    1000000

course00@daint104:~> 
```

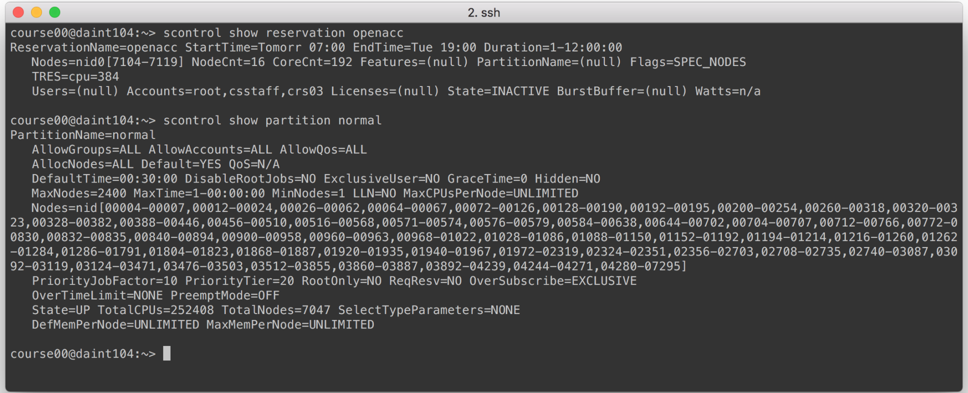
Running on Piz Daint

Other useful commands

- `squeue [OPTIONS]`: Check the status of the system job queue
  - Useful options: `-u [USERNAME]`, `-j [JOBID]`
- `scancel [JOBID]`: Cancel a job
- `scontrol`: Detailed information about partitions, reservations, computing nodes etc.

Running on Piz Daint

Other useful commands



```
course00@daint104:~> scontrol show reservation openacc
ReservationName=openacc StartTime=Tomorr 07:00 EndTime=Tue 19:00 Duration=1-12:00:00
   Nodes=nid0[7104-7119] NodeCnt=16 CoreCnt=192 Features=(null) PartitionName=(null) Flags=SPEC_NODES
   TRES=cpu=384
   Users=(null) Accounts=root,csstaff,crs03 Licenses=(null) State=INACTIVE BurstBuffer=(null) Watts=n/a

course00@daint104:~> scontrol show partition normal
PartitionName=normal
   AllowGroups=ALL AllowAccounts=ALL AllowQos=ALL
   AllocNodes=ALL Default=YES QoS=N/A
   DefaultTime=00:30:00 DisableRootJobs=NO ExclusiveUser=NO GraceTime=0 Hidden=NO
   MaxNodes=2400 MaxTime=1-00:00:00 MinNodes=1 LLN=NO MaxCPUsPerNode=UNLIMITED
   Nodes=nid[00004-00007,00012-00024,00026-00062,00064-00067,00072-00126,00128-00190,00192-00195,00200-00254,00260-00318,00320-003
23,00328-00382,00388-00446,00456-00510,00516-00568,00571-00574,00576-00579,00584-00638,00644-00702,00704-00707,00712-00766,00772-0
0830,00832-00835,00840-00894,00900-00958,00960-00963,00968-01022,01028-01086,01088-01150,01152-01192,01194-01214,01216-01260,01262
-01284,01286-01791,01804-01823,01868-01887,01920-01935,01940-01967,01972-02319,02324-02351,02356-02703,02708-02735,02740-03087,030
92-03119,03124-03471,03476-03503,03512-03855,03860-03887,03892-04239,04244-04271,04280-07295]
   PriorityJobFactor=10 PriorityTier=20 RootOnly=NO ReqResv=NO OverSubscribe=EXCLUSIVE
   OverTimeLimit=NONE PreemptMode=OFF
   State=UP TotalCPUs=252408 TotalNodes=7047 SelectTypeParameters=NONE
   DefMemPerNode=UNLIMITED MaxMemPerNode=UNLIMITED

course00@daint104:~>
```

Editing files

- `vim` or `gvim` (X version)
- `emacs -nw` or just `emacs` (X version)
- `gedit` (X only)

Moving data to/from CSCS

- `scp`: Remote copy over SSH
  - Getting a file: `scp course00@ela.cscs.ch:remotefile localfile`
  - Getting a directory: `scp -r course00@ela.cscs.ch:remotedir localdir`
  - Sending a file: `scp localfile course00@ela.cscs.ch:remotefile`
  - Sending a directory: `scp localdir course00@ela.cscs.ch:remotedir`
- `rsync`: Synchronize files remotely over SSH
  - `rsync -avz course00@ela.cscs.ch:remotedir/ localdir/`
  - `rsync -avz localdir/ course00@ela.cscs.ch:remotedir/`
  - Pay attention to the slashes! *rsync behaves differently with or without slashes.*