CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Introduction to the programming environment

CSCS-USI Summer School 2017
http://github.com/eth-cscs/SummerSchool2017/wiki
July 17, 2017

Summary

- Accessing CSCS
- Compiling my code
- Running my code
- Editing my code
- Transferring files from/to CSCS

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Accessing CSCS

Accessing CSCS: SSH

Secure shell: 2 steps

- Piz Daint is hidden: ela / daint10x / `nidxxx`
- frontend first `Ela`: ssh -Y studNN@ela.cscs.ch
- then login node `Piz Daint`: ssh -Y daint

ssh -X stud51@ela.cscs.ch

```
  ================================================================
              IMPORTANT REMINDER FOR USERS of CSCS facilities
           help@cscs.ch - +41 91 610 82 10 - http://user.cscs.ch
  ================================================================

stud51@ela1:~>
```

ssh -Y daint

```
stud51@ela1:~> ssh -Y daint
stud51@daint101:~> xterm
```

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Compiling the code

Compiling the code: setup your PE

Cray Programming Environment

- 4 compilers available: `CCE`*, `GNU`, `INTEL`, `PGI`
- 4 predefined Programming Environment:
  - PrgEnv-`cray`, PrgEnv-`gnu`, PrgEnv-`intel`, PrgEnv-`pgi`
  - `echo $PE_ENV` to get the current PrgEnv
- 3 wrappers available: `ftn` (Fortran), `cc` (C), `CC` (C++)
  - You **must** use the wrappers to compile for the compute node,
  - The wrappers support serial, OpenMP, MPI and Cuda codes.
- The wrappers are based on the `module` command
  - dynamic modification of the **programming environment** via modulefiles,
  - the wrappers will detect the loaded **modulefiles**,
  - and will automatically add the needed flags and libraries.

Compiling the code: module list

module list

```
*stud51@daint103:~> $ module list
Currently Loaded Modulefiles:
  1) modules/3.2.10.5                      13) job/2.0.2_g98a4850-2.43
  2) eswrap/2.0.11-2.2                     14) dvs/2.7_2.0.72_gb37c41f
       -2.184
 *3) cce/8.5.5                             15) alps/6.2.7-22.3
  4) craype-network-aries                  16) rca/2.0.10_g66b76b7-2.51
  5) craype/2.5.8                          17) atp/2.0.4
 *6) cray-libsci/16.11.1                  *18) PrgEnv-cray/6.0.3
  7) udreg/2.3.2-4.14                     *19) cray-mpich/7.5.0
  8) ugni/6.0.13-2.8                       20) slurm/17.02.3-1
  9) pmi/5.0.10-1.0000.11050.0.0.ari       21) ddt/7.0.2
 10) dmapp/7.1.0-16.18                    *22) craype-haswell
 11) gni-headers/5.0.9-2.7                 23) xalt/daint-2016.11
 12) xpmem/2.0.3_geb8008a-2.11
```

Compiling the code: module swap

module switch from CCE to GNU

```
stud51@daint103:~> module switch PrgEnv-cray PrgEnv-gnu
  Currently Loaded Modulefiles:
  1) modules/3.2.10.5                    13) ugni/6.0.13-2.8
  2) eswrap/2.0.11-2.2                   14) pmi
       /5.0.10-1.0000.11050.0.0.ari
 *3) gcc/5.3.0                           15) dmapp/7.1.0-16.18
...
 11) cray-libsci/16.11.1                *23) PrgEnv-gnu/6.0.3
 12) udreg/2.3.2-4.14

stud51@daint103:~> ftn --version
GNU Fortran (GCC) 5.3.0 20151204 (Cray Inc.)
```

module switch from GNU to CCE

```
stud51@daint103:~> module switch PrgEnv-gnu PrgEnv-cray
Currently Loaded Modulefiles:
...
 *6) cce/8.5.5                           18) job/2.0.2_g98a4850-2.43
...
 11) cray-libsci/16.11.1                *23) PrgEnv-cray/6.0.3
 12) udreg/2.3.2-4.14

stud51@daint103:~> ftn -V
Cray Fortran : Version 8.5.5  Thu Jul 13, 2017  16:08:20
```

Compiling the code: module avail

module avail

```
stud51@daint103:~> module load daint-gpu # <- for non-Cray provided
stud51@daint103:~> module avail

# --- COMPILERS ---
PrgEnv-cray/6.0.3  PrgEnv-intel/6.0.3
PrgEnv-gnu/6.0.3   PrgEnv-pgi/6.0.3
cce/8.5.5 gcc/5.3.0 intel/17.0.1.132 pgi/16.9.0
cray-mpich/7.5.0  *Python/3.5.2-CrayGNU-2016.11

# --- TOOLS ---
ddt/7.0.2
perftools/6.4.3
*Scalasca/2.3.1-CrayGNU-2016.11 *Score-P/3.0-CrayGNU-2016.11

# --- LIBS ---
cray-libsci/16.09.1
cray-hdf5-parallel/1.10.0 cray-netcdf-hdf5parallel/4.4.1
cray-petsc-64/3.7.2.1 cray-tpsl-64/16.07.1 cray-trilinos/12.6.3.3
fftw/3.3.4.10

# --- GPU ---
craype-accel-nvidia60
cudatoolkit/8.0.54_2.2.8_ga620558-2.1
cray-libsci_acc/16.11.1

# --- APPS ---
*CP2K/4.1-CrayGNU-2016.11-cuda-8.0
*QuantumESPRESSO/5.4.0-CrayIntel-2016.11
```

Compiling the code: module show/help

module avail cray-hdf5-parallel

```
stud51@daint103:~> module avail cray-hdf5

------------------------------ /opt/cray/pe/modulefiles ------------
cray-hdf5-parallel/1.10.0(default) cray-hdf5-parallel/1.10.0.1
cray-hdf5-parallel/1.8.16
```

module show cray-hdf5-parallel

```
> module show cray-hdf5-parallel    # CCE
setenv        HDF5_DIR /opt/cray/pe/hdf5-parallel/1.10.0/CRAY/8.3

> module switch PrgEnv-cray PrgEnv-gnu  # GNU
> module show cray-hdf5-parallel       # GNU
setenv        HDF5_DIR /opt/cray/pe/hdf5-parallel/1.10.0/GNU/5.1
```

module help cray-hdf5-parallel

```
stud51@daint103:~> module help cray-hdf5-parallel
-------- Module Specific Help for 'cray-hdf5-parallel/1.10.0' ----
...
```

Compiling the code: module load/rm

module load cray-hdf5-parallel

```
stud51@daint103:~> module load cray-hdf5-parallel
stud51@daint103:~> module list
stud51@daint103:~> which h5dump
/opt/cray/pe/hdf5/1.10.0/bin/h5dump
```

module rm cray-hdf5-parallel

```
stud51@daint103:~> module unload cray-hdf5-parallel
stud51@daint103:~> which h5dump
which: no h5dump in (<long_path>)
```

Compiling the code: mini-app

Get the source

```
stud51@daint103:~> git clone \
  https://github.com/eth-cscs/SummerSchool2017.git
Cloning into 'SummerSchool2017'...
```

Compile the Fortran version

```
stud51@daint103:~> cd SummerSchool2017/miniapp/serial/fortran/
stud51@daint103:~> make clean
rm -f main *.o *.i *.mod output.*

stud51@daint103:~> make
ftn -O3 -fopenmp  -c stats.f90 -o stats.o
ftn -O3 -fopenmp  -c data.f90 -o data.o
ftn -O3 -fopenmp  -c operators.f90 -o operators.o
ftn -O3 -fopenmp  -c linalg.f90 -o linalg.o
ftn -O3 -fopenmp *.o main.f90  -o main
```

Compile the C++ version

```
stud51@daint103:~> cd SummerSchool2016.git/miniapp/serial/cxx/
stud51@daint103:~> make
CC -O3 -fopenmp  -c stats.cpp -o stats.o
CC -O3 -fopenmp  -c data.cpp -o data.o
CC -O3 -fopenmp  -c operators.cpp -o operators.o
CC -O3 -fopenmp  -c linalg.cpp -o linalg.o
CC -O3 -fopenmp *.o main.cpp -o main
```

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Running the code

Running your code (1)

srun

- The job submission system used at CSCS is **native Slurm**
- srun allows you to connect to the compute nodes directly
- srun `-Cgpu`: grants access to the GPU nodes
- srun `--res=summer`: our reservation for the course
- srun `-N`: number of compute nodes (12 cores max per node)
- srun `-n`: Total number of MPI tasks
- srun `-t`: duration of the session in minutes (default is 1h)
- `https://user.cscs.ch/getting_started/running_jobs/`

srun

```
stud51@daint102:~> srun -Cgpu --res=summer -t1 -N2 hostname
srun: job 2308290 queued and waiting for resources
srun: job 2308290 has been allocated resources
nid03508
nid03509
```

Running your code (2)

Other useful Slurm commands

- `squeue` -u $USER : what is the status of my salloc session?
- `scontrol` show job `JOBID`: more details about my session?
- `scancel` `JOBID`: cancel my job

squeue/scontrol/scancel

```
stud51@daint102:~> squeue -u stud51
             JOBID     USER      ACCOUNT  ST      REASON NODES    PRIORITY
             1145201   stud51 courses   PD        None      2       15044
stud51@daint102:~> scontrol show job 1145201
JobId=1145201 Name=bash
   UserId=stud51(22854) GroupId=courses(30340)
   Priority=15044 Nice=0 Account=courses QOS=normal
   JobState=RUNNING Reason=None Dependency=(null)
   ...
   NodeList=nid000[68-69]
   NumNodes=2 NumCPUs=16 CPUs/Task=1 ReqB:S:C:T=0:0:*:*
   Socks/Node=* NtasksPerN:B:S:C=0:0:*:1 CoreSpec=0
   MinCPUsNode=1 MinMemoryCPU=2G MinTmpDiskNode=0
   WorkDir=/users/stud51

 stud51@daint102:~> scancel 1145201
```

Running your code (3)

Running the miniapp (serial version)

```
stud51@daint102:~> srun -Cgpu --res=summer -n 1 ./main 256 256 200
    0.01
============================================================
                        Welcome to mini-stencil!
 version :: Fortran90 serial
 mesh    ::  256 * 256      dx = 3.9215688593685627E-3
 time    ::  200 time steps from 0 ..  1.0000000000000002E-2
============================================================
------------------------------------------------------------
      simulation took  5.2622885461896658  seconds
      13761  conjugate gradient iterations
      2615.0219394495398  CG interations per second
      1328  nonlinear newton iterations
------------------------------------------------------------
Goodbye!
```

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Editing the code

Editing the code

Available text editors

- `vim filename` (X version: `gvim`)
- `emacs -nw filename` (X version: `emacs`)
- X only:
  - `gedit`

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Moving data from/to CSCS

Moving data: scp

scp

- Getting a file: `scp USER@FROM:remotefile localfile`
- Sending a file: `scp localfile USER@TO:remotefile`
- Add the `-rp` flag to scp to copy an entire directory

Getting one or more file(s) from CSCS

```
jg@mylaptop > scp -r stud51@ela.cscs.ch:SummerSchool2017/miniapp/ .
data.cpp                    100%  230    271.3KB/s   00:00
data.h                      100% 2547      3.5MB/s   00:00
...
main                        100%   44KB    2.7MB/s   00:00
README.md                   100%  786    998.2KB/s   00:00
jg@mylaptop > evince miniapp.pdf &
```

Sending one or more file(s) to CSCS

```
jg@mylaptop > scp mycode.c stud51@ela.cscs.ch:relative/path
jg@mylaptop > scp mycode.c stud51@ela.cscs.ch:/absolute/path
```