Building and running codes on MareNostrum-IV and CTE-POWER9

Germán Llort, Lau Mercadal
Barcelona Supercomputing Center

Login to MN and setup environment

```
laptop> ssh -Y <USER>@mn1.bsc.es
                     2
login2> module list  3

Currently Loaded Modules:
  1) intel/2017.4   2) impi/2017.4   3) mkl/2017.4   4) bsc/1.0

login2> cd $HOME/tools-material/src/lulesh-ompi

login2> ls
```

OpenMPI? → module switch impi openmpi

GNU? → module switch intel gcc

Building LULESH on MN

```
login2> vi Makefile
```

```
#default build suggestion of MPI + OPENMP with gcc on Livermore machines
#you might have to change the compiler name

SHELL = /bin/sh
.SUFFIXES: .cc .o

LULESH_EXEC = lulesh2.0

MPI_INC = /opt/local/include/openmpi
MPI_LIB = /opt/local/lib

SERCXX = g++ -DUSE_MPI=0
MPICXX = mpicxx -DUSE_MPI=1
CXX = $(MPICXX)
...
CXXFLAGS = -g -O3 -fopenmp -I. -Wall
LDFLAGS  = -g -O3
```

Remove these: MPI compiler adds proper paths

Change to: mpicxx

Compile with debug

Set OpenMP on/off

Building LULESH on MN

```
login2> make

Building lulesh.cc
Building lulesh-comm.cc
Building lulesh-viz.cc
Building lulesh-util.cc
Building lulesh-init.cc
...
Linking
mpicxx ... -o lulesh2.0
```

Running LULESH on MN

```
login2> vi job_lulesh.sh
```

```
#!/usr/bin/env bash
#SBATCH --job-name=lulesh
#SBATCH --output=%j.out
#SBATCH --error=%j.err
#SBATCH --ntasks=8
#SBATCH --cpus-per-task=1
#SBATCH --exclusive
#SBATCH --time=00:05:00
#SBATCH --reservation=PATCVIHPS

module unload ...
module load   ...

export OMP_NUM_THREADS=1

srun ./lulesh2.0 -i 10 -p -s 65
```

If you changed default modules for compiling, set same here

Job submission

```
login2> sbatch job_lulesh.sh
Submitted batch job 4014332

login2> squeue –i 10

  JOBID PARTITION   NAME     USER ST TIME NODES NODELIST(REASON)
4014332      main lulesh nct00XXX PD 0:00     1 (Priority)
```

PD: Pending
R : Running
CD: Completed

```
...
login2> cat 4014332.out 4014332.err
```

See execution log

Building and running LULESH on CTE-POWER9

- Exactly the same ☺

Using CUDA on CTE-POWER9

```
laptop> ssh -Y <USER>@plogin1.bsc.es
                      2

plogin1> module list

Currently Loaded Modules:
  1) gcc/6.4.0   2) openmpi/3.0.0   3) cuda/9.1   4) bsc/commands

plogin1> cd $HOME/tools-material/src/lulesh-cuda

plogin1> ls
```

Select compiler, runtimes… → module avail, unload, load, switch...

Flags to build with CUDA

```
plogin1> vi Makefile
```

```
MPI_FLAGS = -DUSE_MPI

NVCC    = nvcc
FLAGS   = -arch=sm_70 -g
DFLAGS = $(MPI_FLAGS) -lineinfo
RFLAGS = $(MPI_FLAGS) -O3 -DNDEBUG

...

LINKFLAGS = --cudart shared -lmpi

...
```

CUDA front-end compiler

sm_70 is for Tesla v100 GPU architecture

Remember debug!

Link the CUDA runtime dynamically

```
plogin1> make
```

Running with CUDA on CTE-POWER9

```
plogin1> vi job_lulesh_cuda.sh
```

```
#!/usr/bin/env bash
#SBATCH --job-name=lulesh
#SBATCH --output=%j.out
#SBATCH --error=%j.err
#SBATCH --ntasks=8
#SBATCH --gres=gpu:1
#SBATCH --time=00:05:00
#SBATCH --reservation=PATCVIHPS

module unload ...
module load   ...

srun ./lulesh -s 10
```

Request GPU

Same modules used for compiling (omit for defaults)

Compiling BT-MZ on MN

```
laptop> ssh -Y <USER>@mn1.bsc.es
                     2
                     3

login2> cd $HOME/tools-material/src/NPB3.3-MZ-MPI
login2> cp config/make.def.template config/make.def
login2> vi config/make.def

   (line 49) FFLAGS = -O -g
   (line 95) CFLAGS = -O -g

login2> make BT-MZ CLASS=A NPROCS=8

login
   bt-mz.A.8
```

Benchmark (BT, SP, LU)

Number of processes

Problem size: S, W, A, B, … , F (4x increase)

Running BT-MZ on MN

```
login2> vi job_bt-mz.sh
```

```
#!/usr/bin/env bash
#SBATCH --job-name=bt-mz
#SBATCH --output=%j.out
#SBATCH --error=%j.err
#SBATCH --ntasks=8
#SBATCH --exclusive
#SBATCH --time=00:05:00
#SBATCH --reservation=PATCVIHPS

module unload ...
module load   ...

srun ./bin/bt-mz.A.8
```

Same modules used for compiling (omit for defaults)