Building and running NPB-BT-MZ-MPI on Cori

Christian Feld
Jülich Supercomputing Centre

What is the NPB-BT-MZ-MPI?

- A benchmark from the NAS parallel benchmarks suite
  http://www.nas.nasa.gov/Software/NPB
- MPI version
- Implementation in Fortran
- Solves multiple, independent systems of block tridiagonal (BT) equations
- Represents workloads similar to many flow solver codes (3D Navier-Stokes equations)
- Probably not much unused optimization potential

- We will use this application in all exercises during this workshop.

Properties of NPB-BT-MZ-MPI

- The solution is done for multiple zones (MZ), in a repeated time-step loop
  - After each time-step, the zones have to exchange boundary values
  - Fine-grained parallelism within a zone
  - Coarse-grained parallelism between zones
  - Zones are not all equally sized and need to be distributed in a balanced way
- A larger problem size adds more zones
- Exploits multi-level parallelism
  - Hybrid (OpenMP + MPI) implementation
- Suitable testing application for a wide range of tools and analysis types!

First step: Switch to latest Intel environment

- Use the default Intel environment

```
% module list
Currently Loaded Modulefiles:
 1) modules/3.2.6.7                      13) xpmem/0.1-4.5
  2) nsg/1.2.0                            14) job/1.5.5-3.58
  3) modules/3.2.10.4                     15) dvs/2.5_0.9.0-2.155
  4) intel/16.0.3.210.nersc               16) alps/6.1.3-17.12
  5) craype-network-aries                 17) rca/1.0.0-6.21
  6) craype/2.5.5                         18) atp/2.0.2
  7) cray-libsci/16.06.1                  19) PrgEnv-intel/6.0.3
  8) udreg/2.3.2-4.6                      20) craype-haswell
  9) ugni/6.0.12-2.1                      21) cray-shmem/7.4.0
 10) pmi/5.0.10-1.0000.11050.0.0.ari      22) cray-mpich/7.4.0
 11) dmapp/7.1.0-12.37                    23) darshan/3.0.1
 12) gni-headers/5.0.7-3.1
```

Second step: Building the benchmark

- Copy tutorial sources to your work directory:

```
% cd $SCRATCH
% module load training
% tar xzvf $EXAMPLES/NPB3.3-MZ-MPI.tar.gz
% cd NPB-3.3-MZ-MPI
% ls -F
BT-MZ/  Makefile  README.install   SP-MZ/  common/  jobscript/
LU-MZ/  README    README.tutorial  bin/    config/  sys/
```

Building an NPB-MZ-MPI benchmark

```
% make
  ============================================
  =      NAS PARALLEL BENCHMARKS 3.3         =
  =      MPI+OpenMP Multi-Zone Versions      =
  =      F77                                 =
  ============================================

  To make a NAS multi-zone benchmark type

         make <benchmark-name> CLASS=<class> NPROCS=<nprocs>

  where <benchmark-name> is "bt-mz", "lu-mz", or "sp-mz"
        <class>          is "S", "W", "A" through "F"
        <nprocs>         is number of processes

  [...]

 ****************************************************************
 * Custom build configuration is specified in config/make.def   *
 * Suggested tutorial exercise configuration for Bridges:       *
 *       make bt-mz CLASS=B NPROCS=8                            *
 ****************************************************************
```

- Type "make" for instructions

Building an NPB-MZ-MPI benchmark

```
% make bt-mz CLASS=B NPROCS=8
make[1]: Entering directory `BT-MZ'
make[2]: Entering directory `sys'
cc  -o setparams setparams.c -lm
make[2]: Leaving directory `sys'
../sys/setparams bt-mz 8 B
make[2]: Entering directory `../BT-MZ'
ftn -c  -O3 -openmp       bt.f
                                    […]
ftn -c  -O3 -openmp       mpi_setup.f
cd ../common;  ftn -mmic -c  -O3 -openmp        print_results.f
cd ../common;  ftn -mmic -c  -O3 -openmp        timers.f
ftn  -O3 -openmp   -o ../bin/bt-mz_B.30 bt.o
 initialize.o exact_solution.o exact_rhs.o set_constants.o adi.o
 rhs.o zone_setup.o x_solve.o y_solve.o  exch_qbc.o solve_subs.o
 z_solve.o add.o error.o verify.o mpi_setup.o ../common/print_results.o
 ../common/timers.o
make[2]: Leaving directory `BT-MZ'
Built executable ../bin/bt-mz_B.8
make[1]: Leaving directory `BT-MZ'
```

- Specify the benchmark configuration
  - benchmark name: **bt-mz**, lu-mz, sp-mz
  - the number of MPI processes: NPROCS=**8**
  - the benchmark class (S, W, A, B, C, D, E): CLASS=**B**

Shortcut: % **make suite**

Third step: Run the application

- Change to bin/ directory and copy job script from ../jobscript/cori-p1

```
% cd bin
% cp ../jobscript/cori-p1/reference.sbatch.B.8 .
% less reference.sbatch.B.8
```

- Submit the job

```
% sbatch reference.sbatch.B.8
```

Useful commands

- Check your personal job queue:

```
% squeue -u $USER
```

- Cancel a job:

```
% scancel <job id>
```

NPB-MZ-MPI / BT reference execution

```
% less ref-B.8-<job_id>.out
 NAS Parallel Benchmarks (NPB3.3-MZ-MPI) - BT-MZ MPI+OpenMP Benchmark
 Number of zones:   8 x   8
 Iterations: 200    dt:   0.000300
 Number of active processes:     8

 Use the default load factors with threads
 Total number of threads:     32  (  4.0 threads/process)

 Calculated speedup =     31.52

 Time step    1
 Time step   20
  [...]
 Time step  180
 Time step  200
 Verification Successful

 BT-MZ Benchmark Completed.
 Time in seconds =  5.36
```

- Copy jobscript and launch as a hybrid MPI+OpenMP application
- Reproducible? CPU frequency constant? Turboboost? Pinning?

Hint: save the benchmark output (or note the run time) to be able to refer to it later

Done!

You have successfully built and run the benchmark.