Hands-on / Demo: Building and running NPB-MZ-MPI / BT

Markus Geimer
Jülich Supercomputing Centre

What is NPB-MZ-MPI / BT?

- A benchmark from the NAS parallel benchmarks suite
- MPI + OpenMP version
- Implementation in Fortran 77
- Solves multiple, independent systems of block tridiagonal (BT) equations
- Represents workloads similar to many flow solver codes (3D Navier-Stokes equations)
- Probably not much unexploited optimization potential

Properties of NPB-MZ-MPI / BT

- 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 (MPI + OpenMP) implementation
- Suitable testing application for a wide range of tools and analysis types!

Performance analysis steps

- 0.0 Reference execution for validation

- 1.0 Program instrumentation
- 1.1 Summary measurement collection

- 2.0 Summary experiment scoring
- 2.1 Summary measurement collection with filtering
- 2.2 Filtered summary analysis report examination

- 3.0 Event trace collection
- 3.1 Event trace examination & analysis

First step: Set up the environment

- To set up your environment, load the corresponding modules

```
% module load Intel
% module load IntelMPI
```

- This will use Intel compilers and Intel MPI

Second step: Build the benchmark

- Extract tutorial sources in your work directory

```
% cd $WORK
% mkdir <id>
% cd <id>
% tar xvzf ~train139/Tools/perftools/NPB3.3-MZ-MPI.tar.gz
```

- Enter source directory

```
% cd NPB-3.3-MZ-MPI
% ls -F
BT-MZ/   Makefile    README.install     SP-MZ/    config/      sys/
LU-MZ/   README      README.tutorial    common/   jobscript/
```

- Check build configuration

```
% less config/make.def
```

Second step: Build the benchmark (cont.)

```
% make bt-mz NPROCS=8 CLASS=C
make[1]: Entering directory `BT-MZ'
make[2]: Entering directory `sys'
icc  -o setparams setparams.c -lm
make[2]: Leaving directory `sys'
../sys/setparams bt-mz 8 C
make[2]: Entering directory `../BT-MZ'
mpiifort -c  -O3 -qopenmp       bt.f
 [...]
mpiifort -c  -O3 -qopenmp       mpi_setup.f
cd ../common;  mpiifort -c  -O3 -qopenmp       print_results.f
cd ../common;  mpiifort -c  -O3 -qopenmp       timers.f
mpiifort  -O3 -qopenmp   -o ../bin/bt-mz_C.8 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_C.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=**C**

Third step: Run the application

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

```
% cd bin
% cp ../jobscript/jureca/reference.sbatch .
```

- Check the jobscript

```
% less reference.sbatch
```

- Submit the job

```
% sbatch reference.sbatch
```

NPB-MZ-MPI / BT reference execution

```
% less mzmpibt.o<job_id>
 NAS Parallel Benchmarks (NPB3.3-MZ-MPI) - BT-MZ MPI+OpenMP Benchmark
 Number of zones:  16 x  16
 Iterations:  200   dt:   0.000100
 Number of active processes:     8
 Total number of threads:       48   (  6.0 threads/process)

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

 BT-MZ Benchmark Completed.
 Time in seconds = 15.97
```

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.