Hands-on: *JUSUF*NPB-MZ-MPI / bt-mz_C.8

VI-HPS Team















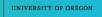














Tutorial exercise objectives

- Familiarise with usage of VI-HPS tools
 - complementary tools' capabilities & interoperability
- Prepare to apply tools productively to your applications(s)
- Exercise is based on a small portable benchmark code
 - unlikely to have significant optimisation opportunities
- Optional (recommended) exercise extensions
 - analyse performance of alternative configurations
 - investigate effectiveness of system-specific compiler/MPI optimisations and/or placement/binding/affinity capabilities
 - investigate scalability and analyse scalability limiters
 - compare performance on different HPC platforms
 - **...**



Environment, Materials, and MPI modules (JUSUF)

Activate the project environment and load the modules

```
% cd /p/project/training/2214
% source setup.sh
```

Copy tutorial sources to your WORK directory

```
% cd work/[your_JUSUF_username]
% cp /p/project/training2214/NPB3.3-MZ-MPI.tar.gz .
% tar -zxvf NPB3.3-MZ-MPI.tar.gz
% cd NPB3.3-MZ-MPI
```

Directory for data exchange during the workshop

```
% /p/project/training2214
```

Environment, Materials, and MPI modules (JUSUF)

Activate the project environment and load the modules

```
% jutil env activate --project training2214 --budget training2214
% module load Stages/2022 # default (but not yet in Xpra)
% module load GCC ParaStationMPI
% export PATH=$PATH:/p/project/training2214/tools/paraver/4.10.0/bin
```

Copy tutorial sources to your HOME directory

```
% cd $HOME
% cp /p/project/training2214/NPB3.3-MZ-MPI.tar.gz .
% tar -zxvf NPB3.3-MZ-MPI.tar.gz
% cd NPB3.3-MZ-MPI
```

Directory for data exchange during the workshop

```
% /p/project/training2214
```



NPB-MZ-MPI Suite

- The NAS Parallel Benchmark suite (MPI+OpenMP version)
 - Available from:

http://www.nas.nasa.gov/Software/NPB

- 3 benchmarks in Fortran77
- Configurable for various sizes & classes
- Move into the NPB3.3-MZ-MPI root directory

```
% ls
bin/ common/ jobscript/ Makefile README.install SP-MZ/
BT-MZ/ config/ LU-MZ/ README README.tutorial sys/
```

- Subdirectories contain source code for each benchmark
 - plus additional configuration and common code
- The provided distribution has already been configured for the tutorial, such that it is ready to "make" one or more of the benchmarks
 - but config/make.def may first need to be adjusted to specify appropriate compiler flags



NPB-MZ-MPI / BT: config/make.def

```
SITE- AND/OR PLATFORM-SPECIFIC DEFINITIONS.
# Configured for generic MPI with GCC compiler
OPENMP = -fopenmp  # GCC compiler

#OPENMP = -qopenmp  # Intel compiler
# The Fortran compiler used for MPI programs
                                                                                   Default (no instrumentation)
MPIF77 = mpif77
# Alternative variants to perform instrumentation
                                                                                  Hint: uncomment a compiler
#MPIF77 = scorep --user mpif77
                                                                                  wrapper to do instrumentation
```



Building an NPB-MZ-MPI Benchmark

```
% make
       NAS PARALLEL BENCHMARKS 3.3
       MPI+OpenMP Multi-Zone Versions
 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"
                    is "S", "W", "A" through "F"
       <class>
       <nprocs>
                     is number of processes
 [...]
       *****************
* Custom build configuration is specified in config/make.def *
* Suggested tutorial exercise configuration for Marenostrum:
        make bt-mz CLASS=C NPROCS=8
```

Type "make" for instructions

Building an NPB-MZ-MPI Benchmark

```
% make bt-mz CLASS=C 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 C
make[2]: Entering directory `../BT-MZ'
mpif77 -q -c -O3 -fopenmp
                                 bt.f
[...]
mpif77 -g -c -O3 -fopenmp mpi setup.f
cd ../common; mpif77 -q -c -O3 -fopenmp print results.f
cd ../common; mpif77 -g -c -O3 -fopenmp timers.f
mpif77 -q -O3 -fopenmp -o ../bin/bt-mz B.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 benchmark class (S, W, A, B, C, D, E): CLASS=**B**
 - the number of MPI processes: NPROCS=8



NPB-MZ-MPI / BT (Block Tridiagonal Solver)

- What does it do?
 - Solves a discretized version of the unsteady, compressible Navier-Stokes equations in three spatial dimensions
 - Performs 200 time-steps on a regular 3-dimensional grid
- Implemented in 20 or so Fortran77 source modules
- Uses MPI & OpenMP in combination
 - 8 processes each with 5 threads for hands-on with Cube
 - bt-mz_C.8 will run for around 30 secs



NPB-MZ-MPI / BT Reference Execution

```
% cd bin
% cp ../jobscript/jusuf/reference.sbatch .
% sbatch reference.sbatch
% cat npb btmz.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:
Use the default load factors with threads
Total number of threads: 48 ( 6.0 threads/process)
Time step
Time step
            20
 [...]
Time step 180
Time step 200
Verification Successful
BT-MZ Benchmark Completed.
Time in seconds = 14.93
```

Copy jobscript and launch as a hybrid MPI+OpenMP application

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