Hands-on: Fujitsu PrimeHPC FX10 *Pi* [Π]
NPB-MZ-MPI / BT

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
  - …

Access to *Pi* FX10

```
% ssh –X -C vihps2016NN@pi.ircpi.kobe-u.ac.jp

[vihps2016NN@pi ~]$ pwd
/home/S11505/vihps2016NN
```

- Logging on to *Pi*
  - *use your provided account vihps2016**NN***
  - *enable X11 forwarding to be able to use graphical tools*

```
% ls /home/S11505/shared
tools/
tutorial/
```

Tutorial materials

- File systems
  - No optimised parallel filesystem available
  - Tutorial materials and tools installed in shared directory
  - Untar tutorial exercise sources in your working directory

```
% cd
% tar zxvf /home/S11505/shared/tutorial/NPB3.3-MZ-MPI.tar.gz
% cd NPB3.3-MZ-MPI
```

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 and install them into a (tool-specific) “bin” subdirectory

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 FX10 systems: *
 *       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'
mpifrtpx -c -Fwide -Kfast -Kopenmp       bt.f
                                  […]
mpifrtpx -c -Fwide -Kfast -Kopenmp       mpi_setup.f
cd ../common;  mpifrtpx -c –Fwide -Kfast -Kopenmp       print_results.f
cd ../common;  mpifrtpx –c –Fwide -Kfast -Kopenmp       timers.f
mpifrtpx  -Kfast -Kopenmp   -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_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**

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 4 threads should be reasonable for 2 compute nodes of *Pi*
  - bt-mz_B.8 should run in around 16 seconds
  - bt-mz_C.8 should take around 65 seconds

NPB-MZ-MPI / BT Reference Execution

```
% cd bin
% cp ../jobscript/fx10/run.sh .
% less run.sh
% pjsub ./run.sh
% cat run.sh.o<job_id>
 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
 Total number of threads:       32  (  4.0 threads/process)

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

 BT-MZ Benchmark Completed.
 Time in seconds = 16.72
```

- 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

Job submission and start

```
% pjsub ./jobscript.sh
```

```
#PJM –j                                  # combine stdout & stderr
#PJM --mpi proc=8                        # number of MPI ranks
#PJM --rsc-list node=2                   # number of compute nodes
#PJM --rsc-list elapse=10:00             # (max) duration min:sec
#PJM --rsc-list rscgrp=small             # resource queue

export OMP_NUM_THREADS=4
export NPROCS = 8

mpiexec –np $NPROCS ./a.out
```

- Submit jobscript with `pjsub`
- Minimal jobscript for MPI+OMP: e.g., 8 MPI ranks each with 4 OMP threads on 2 compute nodes

```
% pjstat
% pjdel <jobid>
```

- View job queue
- Cancel job

Local tools installation (*Pi* Fujitsu FX10)

- Fujitsu (cross-)compilers already on PATH
  - mpifccpx [C], mpiFCCpx [C++], mpifrtpx [Fortran]

- Setup PATH with VI-HPS tools
  - Adds BSC, JSC & UO tools to shell PATH

```
% source /home/S11505/shared/tools/setup.sh
```

  - Hint: add this line to your $HOME/.bashrc

Tutorial Exercise Steps

- Edit config/make.def to adjust build configuration
  - Modify specification of compiler/linker: MPIF77
- Make clean and build new tool-specific executable

```
% make clean
% make bt-mz CLASS=B NPROCS=8
Built executable ../bin.$(TOOL)/bt-mz_B.8
```

- Change to the directory containing the new executable before running it with the desired tool configuration

```
% cd bin.$(TOOL)
% cp ../jobscript/fx10/$(TOOL).sh  .
% pjsub ./$(TOOL).sh
```

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

```
#                SITE- AND/OR PLATFORM-SPECIFIC DEFINITIONS.
#
#---------------------------------------------------------------------------

#---------------------------------------------------------------------------
# Configured for Fujitsu MPI and cross-compilers for FX10/K
#---------------------------------------------------------------------------
OPENMP = -Kopenmp         # Fujitsu compiler

...
#---------------------------------------------------------------------------
# The Fortran compiler used for MPI programs
#---------------------------------------------------------------------------
MPIF77 = mpifrtpx # Fujitsu compiler

# Alternative variant to perform instrumentation
#MPIF77 = scorep --user  mpifrtpx

# PREP is a generic preposition macro for instrumentation preparation
#MPIF77 = $(PREP) mpifrtpx
...
```

Default (no instrumentation)

Hint: uncomment a compiler wrapper to do instrumentation