Hands-on: SGI UltraViolet2 *uv2*
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 *uv2* UltraViolet2

```
% ssh –YAC lxlinux1.lrz.de –l di49XYZ
di49XYZ@lxa191:~> ssh –YAC ice1-login
di49XYZ@ice1-login:~> pwd
/home/hpc/a2c06/di49XYZ
```

```
% ls /home/hpc/a2c06/lu23bud/LRZ-VIHPSTW21
tools/
tutorial/
```

Tutorial materials

- Logging on to *uv2*
  - *use your provided account* ***di49XYZ***
  - *enable X11 forwarding to be able to use graphical tools*
- 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/hpc/a2c06/lu23bud/LRZ-VIHPSTW2/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 HPC systems:  *
 *       make bt-mz CLASS=B NPROCS=4                           *
 ***********************************************************
```

- Type “make” for instructions

Building the NPB-MZ-MPI Benchmark

```
% make bt-mz CLASS=B NPROCS=4
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 4 B
make[2]: Entering directory `../BT-MZ'
mpif77 -c –O3 –openmp –extend-source      bt_scorep_user.F
                                    […]
mpif77 -c –O3 -openmp –extend-source      mpi_setup.f
cd ../common;  mpif77 -c –O3 –openmp –extend-source        print_results.f
cd ../common;  mpif77 -c –O3 –openmp –extend-source        timers.f
mpif77  -O3 -openmp –extend-source  -o ../bin/bt-mz_B.4 bt_scorep_user.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.4
make[1]: Leaving directory `BT-MZ'
```

- Specify the benchmark configuration
  - benchmark name: **bt-mz**, lu-mz, sp-mz
  - the number of MPI processes: NPROCS=**4**
  - 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
  - 4 processes each with 4 threads should be reasonable for *uv2*
  - bt-mz_B.4 should run in around 19 seconds
  - bt-mz_C.4 should take around 75 seconds

NPB-MZ-MPI / BT Reference Execution

```
% cd bin
% cp ../jobscript/lrz_uv2_mpt/run.mpt.sbatch .
% less run.mpt.sbatch
% sbatch ./run.mpt.sbatch
% cat bt-mz.mpt.<job_id>.uv2.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:     4
 Total number of threads:       16  (  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 = 18.99
```

- Copy provided jobscript for UV2 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

```
% sbatch ./jobscript.sbatch
```

```
#!/bin/bash
#SBATCH –-clusters=uv2                # run on UltraViolet2
#SBATCH --ntasks=4                    # number of MPI ranks
#SBATCH –-cpus-per-task=4             # cores reserved per process
#SBATCH --time=00:15:00               # (max) duration hr:min:sec

export OMP_NUM_THREADS=4
export NPROCS=4

srun_ps –n $NPROCS –t $OMP_NUM_THREADS  ./a.out
```

- Submit jobscript with sbatch
- Minimal jobscript for MPI+OMP: e.g., 4 MPI ranks each with 4 OMP threads

```
% squeue --clusters=uv2
% scancel --clusters=uv2 <jobid>
```

- View job queue
- Cancel job

Local tools installation (*uv2* SGI UltraViolet2)

- SGI MPT MPI 2.09 with Intel 15.0 compilers already on PATH
  - mpicc [C], mpiCC [C++], mpif77[Fortran77], mpif90 [Fortran90]

- Setup PATH with VI-HPS tools

```
% source /home/hpc/a2c06/lu23bip/LRZ-VIHPSTW21/tools/source-me.scorep-2.0.1.mpt.sh
```

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

- If you switch modules to Intel MPI or GCC compilers
  - compilers, flags and jobscripts need to be revised
  - different versions of VI-HPS tools are also required

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=4
Built executable ../bin.$(TOOL)/bt-mz_B.4
```

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

```
% cd bin.$(TOOL)
% cp ../jobscript/lrz_uv2_mpt/$(TOOL).sh  .
% sbatch ./$(TOOL).sbatch
```

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

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

#---------------------------------------------------------------------
# Configured for generic MPI
#---------------------------------------------------------------------
#COMPFLAGS = -fopenmp -ffixed-line-length-none # GCC (gfortran) compiler
COMPFLAGS = -openmp -extend-source # Intel (ifort) compiler
...
#---------------------------------------------------------------------
# The Fortran compiler used for MPI programs
#---------------------------------------------------------------------
MPIF77 = mpif77 # generic MPI compiler

# Alternative variant to perform instrumentation
#MPIF77 = scorep --user  mpif77

# PREP is a generic preposition macro for instrumentation preparation
#MPIF77 = $(PREP) mpif77
...
```

Default (no instrumentation)

Hint: uncomment a compiler wrapper to do instrumentation