Virtual Institute – High Productivity Supercomputing

VI-HPS

Hands-on exercise: NPB-OMP / BT

VI-HPS Team

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

German Research School for Simulation Sciences

RWTH Rheinisch-Westfälische Technische Hochschule Aachen

TUM Technische Universität München

Universität Stuttgart

Université de Versailles Saint-Quentin-en-Yvelines

Jülich Forschungszentrum

Lawrence Livermore National Laboratory

Technische Universität Dresden

University of Oregon

The University of Tennessee

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
  - …

Local installation for workshop

- Log-in to cluster with X11 forwarding enabled

```
% ssh –X igloo.calcul.ecp.fr
```

- Copy the tutorial sources to your working directory

```
% cd <myworkdir>
% cp –r /home/wylieb/tutorial/Exercies/NPB3.3-OMP .
% cd NPB3.3-OMP
```

  - (When available, generally advisable to use a parallel filesystem such as $WORK)

NPB-MZ-MPI Suite

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

    http://www.nas.nasa.gov/Software/NPB
  - 8 benchmarks in Fortran77, 2 in C
  - Configurable for various sizes & classes
- Move into the NPB3.3-OMP root directory

```
% ls
bin/     common/  jobscript/  Makefile  README.install   SP/
BT/      config/  LU/         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's ready to “make” one or more of the benchmarks and install them into a (tool-specific) “bin” subdirectory

Building an NPB-MZ-MPI Benchmark

- Type “make” for instructions

```
% make
  ===========================================
  =      NAS PARALLEL BENCHMARKS 3.3        =
  =      OpenMP Versions                    =
  =      F77/C                              =
  ===========================================

  To make a NAS benchmark type

         make <benchmark-name> CLASS=<class>

  where <benchmark-name> is "bt", "cg", "ep", "ft", "is", "lu",
                             "mg", "sp", "ua", or "dc"
         <class>          is "S", "W", "A", "B", "C" or "D"
  [...]

 ***************************************************************
 * Custom build configuration is specified in config/make.def  *
 * Suggested tutorial exercise configuration for HPC systems:  *
 *       make bt CLASS=B                                       *
 ***************************************************************
```

Building an NPB-MZ-MPI Benchmark

- Specify the benchmark configuration
  - benchmark name: **bt**, cg, ep, ft, is, lu, mg, sp, ua
  - the benchmark class (S, W, A, B, C, D, E): CLASS=**B**

```
% make bt CLASS=B
cd BT; make CLASS=B VERSION=
make: Entering directory 'BT'
cd ../sys; cc  -o setparams setparams.c
../sys/setparams bt B
ifort -c  -O –g -openmp bt.f
 [...]
cd ../common;  ifort -c  -O -g -openmp timers.f
ifort –O –g -openmp -o ../bin/bt_B \
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
Built executable ../bin/bt_B
make: Leaving directory 'BT'
```

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 OpenMP
  - on ICE with 12 threads, class B takes around 30 seconds

NPB-MZ-MPI / BT Reference Execution

- *In the subdirectory with the executable,* copy provided jobscript and launch the OpenMP application

```
% cd bin
% cp ../jobscript/intel/run.pbs .
% less run.pbs
% qsub run.pbs
% cat run.o<id>
 NAS Parallel Benchmarks (NPB3.3-OMP) - BT OpenMP Benchmark
 Size:  102x 102x 102
 Iterations:  200   dt:   0.000300
 Number of available threads:    12

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

 BT Benchmark Completed.
 Time in seconds = 30.25
```

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

NPB-OMP / BT: config/make.def (Intel version)

```
#            SITE- AND/OR PLATFORM-SPECIFIC DEFINITIONS
#---------------------------------------------------------------------
# Items in this file may need to be changed for each platform.
#---------------------------------------------------------------------

# Configured for Intel compilers
OPENMP = -openmp

#---------------------------------------------------------------------
# The Fortran compiler used for OpenMP programs
#--------------------------------------------
F77 = ifort

#F77 = scorep  ifort

# PREP is a generic preposition macro for ins
#F77 = $(PREP) ifort

# This links OMP Fortran programs; usually the same as ${F77}
FLINK   = $(F77)
...
```

Default (no instrumentation)

Hint: uncomment one of these alternative compiler wrappers to perform instrumentation

NPB-OMP / BT: config/make.def (GCC version)

```
#            SITE- AND/OR PLATFORM-SPECIFIC DEFINITIONS
#---------------------------------------------------------------------
# Items in this file may need to be changed for each platform.
#---------------------------------------------------------------------

# Configured for GCC compilers
OPENMP = -fopenmp

#---------------------------------------------------------------------
# The Fortran compiler used for OpenMP programs
#--------------------------------------------
F77 = gfortran

#F77 = scorep  gfortran

# PREP is a generic preposition macro for ins
#F77 = $(PREP) gfortran

# This links OMP Fortran programs; usually the same as ${F77}
FLINK   = $(F77)
...
```

Default (no instrumentation)

Hint: uncomment one of these alternative compiler wrappers to perform instrumentation