

VI-HPS

SOFTWARE



0.00 <<time step loop>>
0.00 updatedt
6.62 updatex
372.85 updateien
0.00 gene
0.00 <<iteration loop>>
293.65 genbc



PRODUCTIVITY

FAST SOLUTIONS

☒ PAPI_L1_DCM
☒ PAPI_L1_ICM
☐ PAPI_L2_DCM
☒ PAPI_L2_ICM
☒ PAPI_L2_TCM
☐ PAPI_L2_TCM

Hands-on exercise: NPB-MZ-MPI / BT

VI-HPS Team

- 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
 - ...

- Copy tutorial sources to your working directory

```
% cp -r /project/csstaff/courses/CSCS_USI_School/Day8/tutorial .  
% cd tutorial/NPB3.3-MZ-MPI
```

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

- 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's ready to “make” one or more of the benchmarks and install them into a (tool-specific) “bin” subdirectory

- Type “make” for instructions

```
% 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                          *
*****
```

- 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**

```
% make bt-mz CLASS=B NPROCS=4
cd BT-MZ; make CLASS=B NPROCS=4 VERSION=
make: Entering directory 'BT-MZ'
cd ../sys; cc -o setparams setparams.c
../sys/setparams bt-mz 4 B
ftn -c -O3 -openmp bt.f
[...]
cd ../common; ftn -c -O3 -openmp timers.f
ftn -O3 -openmp -o ../bin/bt-mz_B.4 \
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-mz_B.4
make: Leaving directory 'BT-MZ'
```

- What does it do?
 - Solves a discretized version of 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 with 4 threads each should be reasonable
 - bt-mz_B.4 should run in around 30 seconds
 - bt-mz_C.4 should take around 3-4x longer

- Acquire partition and launch as a hybrid MPI+OpenMP application

```
% cd bin
% salloc -N 1
salloc: granted job allocation
% export OMP_NUM_THREADS=4
% aprun -n 4 -d 4 ./bt-mz_B.4
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 = 29.87
```

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

- Copy jobscript and launch as a hybrid MPI+OpenMP application

```
% cd bin
% cp ../jobscript/todi/run.sbatch .
% less run.sbatch
% sbatch run.sbatch
% cat slurm-<id>.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 = 29.87
```

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

- Currently installed in a non-default location
 - Need to extend the module search path manually

```
% module use \  
    /project/csstaff/courses/CSCS_USI_School/Day9/modulefiles  
% module avail  
-- /project/csstaff/courses/CSCS_USI_School/Day9/modulefiles  
scalasca/1.4.3  
scalasca/2.0-rc1 (default)  
scorep/1.1.1  
scorep/1.1.1-cuda (default)  
[...]  
% module load scorep scalasca
```

- NB: use older PrgEnv-gnu environment

```
% module switch PrgEnv-cray PrgEnv-gnu  
% module switch cray-mpich cray-mpich2/5.6.5  
% module switch gcc gcc/4.7.2  
% module switch cray-libsci cray-libsci/12.0.01
```