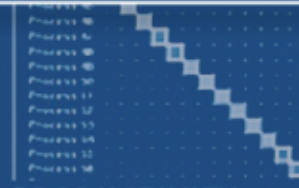




SOFTWARE

+ 19.56 updatex
+ 399.70 updateien
+ 0.00 gene
- 0.00 <<iteration loop>>
+ 447.52 genbc

PRODUCTIVITY



FAST SOLUTIONS

- PAPI_L1_ICM
- PAPI_L2_DCM
- PAPI_L2_ICM
- PAPI_L1_TCM

NPB3.3-MPI/BT tutorial example application

Brian Wylie
Jülich Supercomputing Centre
b.wylie@fz-juelich.de
March 2010

- The NAS Parallel Benchmark suite (sample MPI version)
 - Available from <http://www.nas.nasa.gov/Software/NPB>
 - 9 benchmarks (7 in Fortran77, 2 in C)
 - Configurable for various sizes & classes
- Move into the NPB3.3-MPI root directory

```
% cd workshop-vihps/NPB3.3-MPI; ls
BT/      CG/      DT/      EP/      FT/      IS/      LU/      MG/      SP/
bin/     common/ config/  Makefile  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” benchmarks and install them into a (tool-specific) “bin” subdirectory

- Type “make” for instructions

```
% make
```

```
=====
=      NAS Parallel Benchmarks 3.3      =
=      MPI/F77/C                        =
=====
```

To make a NAS benchmark type

```
make <benchmark-name> NPROCS=<number> CLASS=<class>
```

To make a set of benchmarks, create the file config/suite.def according to the instructions in config/suite.def.template and type

```
make suite
```

```
*****
* Custom build configuration is specified in config/make.def *
* Suggested tutorial benchmark specification:                 *
*   make bt CLASS=W NPROCS=16                                *
*****
```

- Specify the benchmark configuration
 - benchmark name: **bt**, cg, dt, ep, ft, is, lu, mg, sp
 - the number of MPI processes: **NPROC=16**
 - the benchmark class (S, W, A, B, C, D, E): **CLASS=W**

```
% make bt NPROCS=16 CLASS=W
cd BT; make NPROCS=16 CLASS=W SUBTYPE= VERSION=
gmake: Entering directory 'BT'
cd ../sys; cc -o setparams setparams.c
../sys/setparams bt 16 W
mpif77 -c -o bt.f
...
mpif77 -c -o setup_mpi.f
cd ../common; mpif77 -c -o print_results.f
cd ../common; mpif77 -c -o timers.f
mpif77 -c -o btio.f
mpif77 -o -o ../bin/bt_W.16 \
bt.o make_set.o initialize.o exact_solution.o exact_rhs.o \
set_constants.o adi.o define.o copy_faces.o rhs.o solve_subs.o \
x_solve.o y_solve.o z_solve.o add.o error.o verify.o setup_mpi.o \
../common/print_results.o ../common/timers.o btio.o
Built executable ../bin/bt_W.16
gmake: Leaving directory 'BT'
```

- 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
- Can be configured to include various forms of parallel I/O
 - e.g., MPI collective file I/O: SUBTYPE=full
- Implemented in 20 or so Fortran77 source modules
- Needs a square number of processes
 - 16 should be reasonable (decrease to 9 or 4, if necessary)
 - ▶ excess processes idle when run with more than compiled
 - ▶ don't expect to see speed-up when run on a notebook computer!
 - bt_W should run in around 5 to 12 seconds
 - bt_A should take around 16-20x longer (90-100 seconds)

- Launch as an MPI application

```
% cd bin; mpiexec -np 16 ./bt_W.16
NAS Parallel Benchmarks 3.3 -- BT Benchmark
Size: 24x 24x 24
Iterations: 200 dt: 0.0008000
Number of active processes: 16

Time step 1
Time step 20
Time step 40
Time step 60
Time step 80
Time step 100
Time step 120
Time step 140
Time step 160
Time step 180
Time step 200
Verification Successful

BT Benchmark Completed.
Time in seconds = 4.70
```

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

- The tutorial steps are similar and repeated for each tool
- Use the provided NPB3.3-MPI tutorial directory

```
% cd workshop-vihps/NPB3.3-MPI; ls
BT/      CG/      DT/      EP/      FT/      IS/      LU/      MG/      SP/
bin/     common/ config/  Makefile  README  README.tutorial sys/
```

- 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 CLASS=W NPROCS=16
Built executable ../bin.$(TOOL)/bt_W.16
```

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

```
% cd bin.$(TOOL)
% export ...
% mpiexec -np 16 bt_W.16
```

- config/make.def

```
#           SITE- AND/OR PLATFORM-SPECIFIC DEFINITIONS
#-----
# Items in this file may need to be changed for each platform.
...
#-----
# The Fortran compiler used for MPI programs
#-----
MPIF77 = mpif77
# Alternative variants to perform instrumentation
#MPIF77 = marmotf77
#MPIF77 = tau_f90.sh
#MPIF77 = scalasca -instrument mpif77
#MPIF77 = vtf77 -vt:f77 mpif77
#MPIF77 = psc_instrument mpif77 -I/usr/local/packages/openmpi/include

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

# This links MPI Fortran programs; usually the same as ${MPIF77}
FLINK   = $(MPIF77)
...

```

Default (no instrumentation)

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