

Building and running NPB-BT-MZ-MPI on Darwin

Brian Wylie
Jülich Supercomputing Centre

What is the NPB-BT-MZ-MPI?

- A benchmark from the NAS parallel benchmarks suite
- MPI version
- Implementation in Fortran
- Solves multiple, independent systems of block tridiagonal (BT) equations
- Represents workloads similar to many flow solver codes (3D Navier-Stokes equations)
- Probably not much unused optimization potential

- We will use this application in most exercises during this workshop.

Properties of NPB-BT-MZ-MPI

- The solution is done for multiple zones (MZ), in a repeated time-step loop
 - After each time-step, the zones have to exchange boundary values
 - Fine-grained parallelism within a zone
 - Coarse-grained parallelism between zones
 - Zones are not all equally sized and need to be distributed in a balanced way
- A larger problem size adds more zones
- Exploits multi-level parallelism
 - Hybrid (OpenMP + MPI) implementation
- Suitable testing application for a wide range of tools and analysis types!

First step: Switch to latest Intel environment

- Load Intel environment with latest MPI and compilers

```
% module purge  
% module load default-impi-LATEST
```

```
% module list  
Currently Loaded Modulefiles:  
 1) dot  
 2) scheduler  
 3) java/jdk1.8.0_45  
 4) turbovnc/2.0.1  
 5) vgl/2.3.1/64  
 6) intel/impi/5.1.3.181  
 7) global  
 8) intel/cce/16.0.2.181  
 9) intel/fce/16.0.2.181  
10) intel/mkl/11.3.2.181  
11) default-impi-LATEST
```

Second step: Building the benchmark

- Copy tutorial sources to your work directory:

```
% cd $HOME/scratch
% cp -r /home/hpcwyl11/tutorial/NPB3.3-MZ-MPI.tar.gz .
% tar xvzf NPB3.3-MZ-MPI.tar.gz
```

- Create default config/make.def:

```
% cd NPB-3.3-MZ-MPI
% ls -F
BT-MZ/    Makefile    README.install    SP-MZ/    config/    sys/
LU-MZ/    README     README.tutorial   common/    jobscript/
```

- Issue make command (typing only make will give you a help text):

```
% make bt-mz CLASS=B NPROCS=8
```

Third step: Run the application

- Change to bin/ directory and copy job script from ../jobscript/darwin

```
% cd bin  
% cp ../jobscript/darwin/reference.slurm .
```

- Submit the job

```
% sbatch reference.slurm
```

Useful commands

- Check your personal job queue:

```
% squeue -u $USER
```

- Cancel a job:

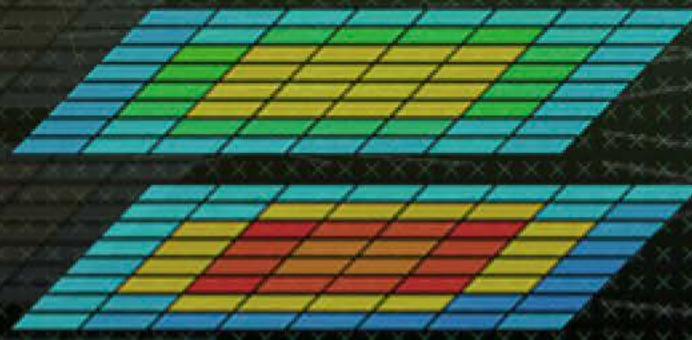
```
% scancel <job id>
```

- Print contents of output file:

```
% cat out.txt
```

- Follow the output, while job is running:

```
% tail -F out.txt
```



Done!

You have successfully built and run the benchmark.