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

Christian Feld
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 GCC (or Intel) environment with latest MPI and compilers

```
% use gcc-4.9.3p mvapich2-gnu-2.2
Prepending: gcc-4.9.3p (ok) mvapich2-gnu-2.2 (ok)
```

Second step: Building the benchmark

- Copy tutorial sources to your work directory:

```
% cd $HOME/scratch
% cp –r /scratch/performance-workshop/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=4
```

Third step: Run the application

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

```
% cd bin
% cp ../jobscript/llnl-cab/reference.sbatch.B.4 .
```

- Submit the job

```
% sbatch reference.sbatch
```

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.

allinea · JÜLICH FORSCHUNGSZENTRUM · RWTH AACHEN UNIVERSITY · TECHNISCHE UNIVERSITÄT DRESDEN · UNIVERSITÉ DE VERSAILLES ST-QUENTIN-EN-YVELINES · Universität Stuttgart · Barcelona Supercomputing Center Centro Nacional de Supercomputación · Lawrence Livermore National Laboratory · TECHNISCHE UNIVERSITÄT DARMSTADT · Technische Universität München · UNIVERSITY OF OREGON · THE UNIVERSITY of TENNESSEE