Building and running NPB-MZ-MPI on Leftraru

Michael Firbach
Technische Universität München

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: Use Intel tools

Add to your **.bashrc**:

```
module load intel impi
```

Re-login to apply changes.

Second step: Building the benchmark

- Copy to your home:

```
$ cp -r /home/courses/instructor06/NPB3.3-MZ-MPI ~
```

This is the original source code, with no modifications.

- Create default **config/make.def**:

```
$ cd NPB3.3-MZ-MPI
$ cp config/make.def.template config/make.def
```

- Edit **config/make.def** to use **mpiifort** as compiler:

```
F77 = mpiifort
```

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

```
$ make bt-mz CLASS=B NPROCS=32
```

Third step: Run the application

Change to **bin** and copy job script from **/home/courses/instructor06/jobscript.slurm**:

```
#!/bin/bash
#SBATCH --job-name=example
#SBATCH --partition=slims
#SBATCH --exclusive=user
#SBATCH --ntasks_per_socket=8
#SBATCH -n 32
#SBATCH --output=out.txt
#SBATCH --error=out.txt
. /etc/profile
. /etc/profile.d/modules.sh
srun ./bt-mz.B.32
```

Run the job with **sbatch <your_script>**

Useful commands

Check your personal job queue:

```
squeue
```

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.