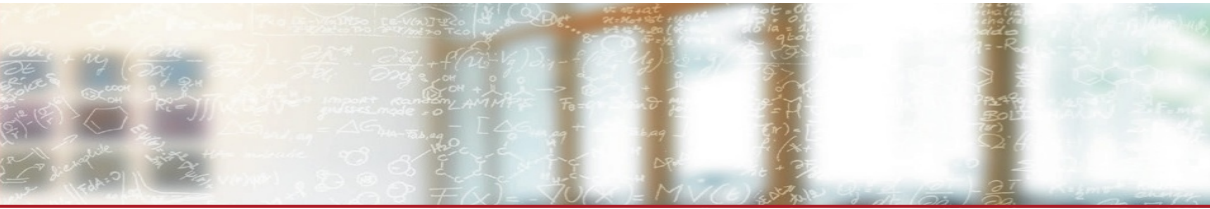
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Hands-on exercices

Directive Based GPU Programming: OpenACC and OpenMP

Vasileios Karakasis, CSCS

29–31 May 2017

Hands-on exercises

General info

- Available at https://github.com/vkarak/openacc-training
  - Clone the repository:
    ```
    git clone https://github.com/vkarak/openacc-training.git
    ```
- `exercises/` → hands-on exercises
- `solutions/` → solutions to exercises (will appear after the course)
- `slides/` → slides of the course

- Get the latest updates with `git pull origin master`

Hands-on exercises

General info

1. `exercises/cray/Himeno`
   - Himeno benchmark
   - Basic directive concepts for accelerators
2. `exercises/openacc/diffusion`
   - 2D diffusion example
   - Interoperability with MPI and CUDA
3. `exercises/openacc/gemm`
   - GEMM
   - Interoperability with CUBLAS

Search for `TODO` for places to modify

Hands-on exercises

How to compile

- Cray compiler
  - Make sure that `PrgEnv-cray` module is loaded
  - `module load craype-accel-nvidia60`
  - `make CRAY=1` (default)
- PGI compiler
  - Make sure that `PrgEnv-pgi` module is loaded
  - `module load cudatoolkit`
  - `make PGI=1`

Hands-on exercises

How to run

1. Source some necessary Slurm presets
   - `source <prefix>/tools/course.sh`
2. Launch directly the job from the command line
   - `srun -Cgpu -N<nnodes> -n<ntasks> <exec> <args> ...`
   - `OMP_NUM_THREADS=<nthreads> srun -Cgpu ...`
3. …or use the job submission script provided
   - `sbatch job.batch`
   - The output is written in `job.out`

2D diffusion example

Source code

- `diffusion2d_omp.{cpp,f90}`
  - Single node OpenMP version for the CPU
- `diffusion2d_openacc.{cpp,f90}`
  - Single node OpenACC version
- `diffusion2d_openacc_mpi.{cpp,f90}`
  - MPI+OpenACC version
  - If `OPENACC_DATA` is undefined, data management is performed by CUDA

2D diffusion example

Executables

Five versions in both C++/Fortran

- `diffusion2d.omp`: OpenMP version for the CPU (baseline)
- `diffusion2d.openacc`: Single-node OpenACC version
- `diffusion2d.openacc.cuda`: Single-node OpenACC version with memory managed by CUDA
- `diffusion2d.openacc.mpi`: MPI+OpenACC version
- `diffusion2d.openacc.cuda.mpi`: MPI+OpenACC version with memory managed by CUDA

- Run as `./<diffusion-exec> <array-dim-pow-2>`

2D diffusion example

Goal of the exercise

- Familiarize with data management in OpenACC
- Learn how to interact with CUDA pointers from OpenACC
- Learn how to enable the fast RDMA data path with Cray MPI

Plot the result with `plot.sh` *and fill in the performance table in* `performance.text`

GEMM

- C++ only (apologies to Fortran guys!)
- Versions
  - Naive: directly from the math textbook (`make CPPFLAGS=''` to enable)
  - Loop reordering with OpenMP on the CPU
  - Loop reordering with OpenACC on the GPU
  - OpenACC + CUBLAS