PRACE Winter School 2017 GPU/KNL Course Cheat Sheet

JuRoPA3

Logging in

ssh train0??@juropa3.zam.kfa-juelich.de, where ?? is the number assigned to you.

Setting up the environment

module use /usr/local/software/juropa3/OtherStages
module load Stages/2016b
module load Intel tbb

Submit a job

sbatch job.sh

A few commands

cd <dir> - switches the working directory to <dir>
ls - lists files in current directory
ls -l - same as ls but gives more detail
mkdir -p <dir1>/<dir2> - creates a new (subdirectory> <dir1>/<dir2>
rm <file> - deletes (removes) a file. Cannot be undone!
less <file> - shows the context of a file.

JURECA

Logging in:

ssh train0??@jureca.fz-juelich.de, where ?? is the number assigned to you.

Setting up the environment

module load Intel ParaStationMPI SciPy-Stack/2016b-Python-2.7.12 freeglut/.3.0.0 CUDA

Accessing a compute node

The frontend nodes can be used for development but to run your code on a GPU you need to access one of the compute nodes. To get a node with four GPUs for 4 hours use
salloc --reservation=prace -p gpus --nodes=1 --gres=gpu:4 --time=2:0:0
After some time you are returned to the prompt. To run a program on the compute node start it with
srun <executable>
To start an interactive session use
srun --forward-x --cpu_bind=none --pty /bin/bash -i

Allocate memory on device

cudaMallocManaged(T** pointer, size_t nbytes)
cudaMalloc(T** pointer, size_t nbytes)

Free memory on device

cudaFree(pointer)

Transfer data

cudaMemcpy(void* dst, void* src, size_t nbytes, enum cudaMemcpyKind dir)
Directions can be one of
cudaMemcpyHostToDevice, cudaMemcpyDeviceToHost, cudaMemcpyDeviceToDevice, cudaMemcpyDefault

Call kernel

kernel_name<<<dim3 grid, dim3 block>>>([args])

CUDA kernel

__global__ void kernel_name([args]) {...} - kernel function
__device__ void device_function([args]) {...} - function that can be called from kernel
__shared__ - fast on-chip memory
All kernel functions have access to
dim3 gridDim, blockDim, blockIdx, threadIdx

CUDA Documentation

You can find lots of documentation at the Nvidia web page: http://docs.nvidia.com/cuda/index.html

Jan H. Meinke <j.meinke@fz-juelich.de>, Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, Jülich, Germany