How to connect to the system

```
ssh -X z3XX@login-gfx2.hpc.cam.ac.uk
```

If you want to use Windows please download **MobaXterm**
(http://mobaxterm.mobatek.net)

Access all material

https://github.com/RSE-Cambridge/2017-cuda-fortran-material

(short url: **https://tinyurl.com/CUDAFORTRAN**)

```
git clone https://github.com/RSE-
Cambridge/2017-cuda-fortran-material.git
```

Environment HW/SW

HW (Wilkes, Cambridge)

- dual socket 6-core Ivy Bridge (12 cores total)
- 2 NVIDIA GPU K20

SW

- PGI 16.10 (community edition)
- MKL 11.3
- Open MPI 1.10.3
- CUDA 8.0
- Few custom libraries (for QE-GPU)

Compile and link using PGI

FORTRAN: pgfortran

C: pgcc

C++: pgc++

Important flags:

- “`-Mpreprocessor`” = enable preprocessor
- “`-mp`” = enable OpenMP
- “`-Mcuda=cc35,cuda8.0`” = enable CUDA FORTRAN
- ”`-Mcudalib=cufft,cublas` ” = link CUBLAS and CUFFT
- “`-pgf90libs`” = runtime libraries for mixed prog lang

SLURM commands

Three command to learn …

```
sbatch <script>

squeue -u $USER

scancel <jobid>
```

Submission scripts

- `submit-monday.sh`
- `submit-tuesday.sh`
- `submit-wednesday.sh`

- `submit-serial.sh`
- `submit.sh`

Example submission script

```
#!/bin/bash
#SBATCH -J NAME
#SBATCH -A TRAINING-GPU
#SBATCH --nodes=UUUUU
#SBATCH --ntasks=XXXXX
#SBATCH --time=0:05:00
#SBATCH --no-requeue
#SBATCH --partition=tesla
#SBATCH --reservation=cuda_fortran_mon

# EDIT ME ONLY IF YOU KNOW WHAT YOU ARE DOING ###############
. /etc/profile.d/modules.sh
module purge
module load default-wilkes
module unload cuda intel/cce intel/fce intel/impi intel/mkl
module load intel/mkl/11.3.3.210
module load pgi/16.10
module load openmpi/pgi/1.10.3
module load cuda/8.0
module load custom/magma/2.0.2 custom/lib-jdr/1.0
#############################################################

cd $SLURM_SUBMIT_DIR

export EXE="./pw.x”
export PARAMS="-input pw.in”

export OMP_NUM_THREADS=6
mpirun -np XXXXX -report-bindings -bind-to socket --map-by ppr:1:socket ${EXE} ${PARAMS}
```

Compile and run QE-GPU code

Two examples: “PW-AUSURF54” & “PW-AUSURF112” (k and gamma)

Compile

```
cd qe-gpu_mini
./configure --enable-parallel --enable-openmp –without-scalapack
cp make.inc-STELLA make.inc
make pw
```

(executable located in “PW/src/pw.x”)

Run

```
export EXE="./pw.x”
export PARAMS="-input ausurf_k.in”

export OMP_NUM_THREADS=6
mpirun -np XXXXX -report-bindings \
  -bind-to socket --map-by ppr:1:socket ${EXE} ${PARAMS}
```