Using OpenMP on Delta

Logging in and environment To run the exercises, you'll need to log into the `Delta` system, with a command like

```
ssh -X s123456@delta.central.cranfield.ac.uk
```

where you need to substitute your account in place of `s123456`.

Computational jobs on `Delta` must be submitted through the queueing system; see the `cuda.sub` sample script.

The system supports two compilers capable of interpreting OpenMP directives: the Intel compiler and the GNU compiler.

GNU Compiler To use the GNU compiler, which will also be used for the CUDA section of the course, you have to set up the appropriate environment variables with the commands

```
module load foss/2016b
```

The `foss/2016b` module loads GCC 5.4.0 which is compatible with CUDA. These commands must be issued every time you log into the system (i.e. for every session). To compile a source file you will need the -fopenmp option, as in the following:

```
gcc -fopenmp -O4 -o hello hello.c
```

Many of the source files also require the -std=c99 option.

Intel Compiler To use the Intel compiler you have to set up the appropriate environment variables with the commands

```
module load intel/2017.03
```

This commands must be issued every time you log into the system (i.e. for every session). To compile a source file you will need the -fopenmp option, as in the following:

```
icc -qopenmp -O3 -o hello hello.c
```

Many of the source files also require the -std=c99 option.

Job submission See the example job subission files `openmp_add.sub` and similar in the source directories; change the email address your Cranfield email; then submit a batch job with the command

```
qsub -V openmp_add.sub
```

If everything has been set up properly, the system will respond with a message like

```
35291.mgmt01
```

(with a new number assigned to each invocation).