Exercise #3: OpenCL matrix multiplication

The goal of this exercise is to practice with OpenCL by implementing a kernel for matrix multiplication.

Since the NVidia module should be already available (see previous exercise), we want to add another platform via ‘module load AMD-APP-SDK’. This adds the AMD CPU-only OpenCL implementation as an additional platform to choose during the context creation.

Download or copy to your home directory the “opencl” directory that you can find in the “exercises” section of the course portal.

Have a look at the code to understand the various operations required to setup OpenCL.

As we did in the previous exercise, instrument the code to measure the time required to fill buffers and run kernels.

How can you tune the grid size (global and local work groups)? How can you procedurally retrieve the maximum values supported? Change both the matrix dimensions and the workgroups to see how far you can push this example (suggestion: comment out the matrix printing part…).

The matrix sizes are defined through two #define and passed to the kernel as arguments. How can you integrate sizes as built-in values in the kernel?

Modify the code to let the user specify the platform to use from the command-line (always default to the first device found, as it is now).

Give a look at Ugo Varetto’s matrix multiplication example for an insight of an optimized implementation using local memory.