CUBLAS

In this exercise, we use the BLAS3 routine DGEMM in cuBLAS to perform a matrix-matrix multiplication using double precision. GEMM is defined as

$$C = \alpha AB + \beta C.$$

where A,B, and C are matrices and α and β are scalars. In our case, we only want to multiply *A* and *B* so we set $\beta = 0$.

The program (dgemm_um.cpp) allocates space for two square matrices *A* and *B* and fills them with random numbers. The data is copied to the GPU on request as soon as DGEMM is invoked; the result is returned back to the host.

Todo

Implement the call to `cublasDgemm` (marked with *TODO*). See the cuBLAS documentation for further information: http://docs.nvidia.com/cuda/cublas/.

Compile the code by calling `make [dgemm_um]`. Run the code on JURON with `make run`. Copy the `bsub` command and change the argument to the program to run the multiplication for different matrix sizes. Try a few!

Is the performance what you would have expected? Compare with the results from the previous exercise.

To convert the time *t*, needed to calculate the matrix product of two matrices of size *n*, into GFLOP/s use

$$\mathrm{GFLOP/s} = 2 * n * n * n / t * 10^{-9}$$

Read the code and find the CUDA specific calls.

Extra credit

Replace random number generator with call to CURAND and remove unnecessary code.