The overall goal of this homework is to obtain comprehensive understanding of the CSR sparse matrix format product, and gain hands-on experience on implementing GPU kernels. The assignment is to program a GPU kernel that converts a dense matrix containing many zero elements into a sparse matrix in CSR format, use the CSR-SPMV from the cuSPARSE library, and compare the runtime with gemv routine.

1. In gpukernels.zip, you find homework.cu, which gives you the framework for the matrix-vector multiplication. A dense matrix is initialized with zeros, and some randomly-chosen non-zero entries. In a first step, add a kernel computing a dense matrix vector product and measure the runtime.

2. Write a GPU kernel that converts the dense matrix into a CSR structure. This may require three steps:
   1. Count the non-zero elements.
   2. Allocate memory for the CSR structure.
   3. Insert the data into the CSR structure.

Your kernel has to be correct, but should also run fast!

3. Use the sparse matrix-vector multiplication routine from NVIDIA’s cuSPARSE library to compute the sparse matrix-vector product using the generated CSR structure. It should pass the correctness check, otherwise your conversion kernel is incorrect. Compare the runtime including the conversion time with the gemv routine, relate it to the number of non-zero elements.

Send me the final version of your program (hanzt@icl.utk.edu) by 20th of April. I expect to compile and run without problems, and to pass the correctness check. Send me also a report including the runtime comparison between gemv and CSR-conversion+SpMV. You should choose large matrix dimensions for this comparison.

Resources:

In the gpukernels.zip you also find the kernels we covered in class. These can help you when writing new kernels.

cuBLAS reference:  http://docs.nvidia.com/cuda/cublas/

cuSPARSE reference:  http://docs.nvidia.com/cuda/cusparse/

In case of difficulties with understanding the assignment, writing GPU kernels, compiling or running the code, get in touch with me, I can help!