General Purpose GPU Computing
CUDA Lecture Rainer Spurzem
Next Lectures: Expected January/Feb. 2013

Exercises:

1. hello - first kernel call, hello world
2. add - vector addition using one thread in one block only
3. add-index - vector addition using blocks in parallel, one thread per block only.
4. add-parallel - vector addition using all blocks and threads in parallel
5. dot - scalar product using shared memory of one block only for reduction
6. dot-full - scalar product using shared memory and atomic add across blocks
7. histo - histogram using fat threads and atomic add on shared and global memory
8. dot-perfect - scalar product using fat threads, shared memory, final reduction on host.

What we will learn from CUDA C:

threadId.x , blockId.x, blockDim.x, gridDim.x	Threads, Blocks
kernel<<<n,m>>> (...)	kernel calls
__device__	device code
__shared__	shared memory on GPU
cudaMalloc / cudaFree	manage global memory of GPU
cudaMemcpy / cudaMemset	copy/set to or from memory
cudaGetDeviceProperties	get device properties in program
cudaEventCreate, cudaEventRecord, cudaEventSynchronize, cudaEventElapsedTime, cudaEventDestroy	CUDA profiling

What we probably do not yet learn in the first lectures:

threadId.y, blockId.y, blockdim.y, gridDim.y	work with 2D grids
__constant__	constant memory on GPU
cudaBindTexture	using texture memory
fat threads for 2D and 3D stencils	thread coalescence optimisation
cudaStreamCreate, cudaStreamDestroy	working with CUDA streams

This lecture has been inspired by Jason Sanders, Introduction to GPU at the GTC2010, and by the book CUDA by Example of Jason Sanders and Edward Kandrot and lectures of Wen-Mei Hwu at the ICCS Workshop and Tutorial in January 2011. Some material from these sources has been used to help in creating the example codes. See

http://silk0.bao.ac.cn/cuda-tutorial/cuda.html

for more information.

北京, November 2012

Rainer Spurzem