OpenCL

CSCI 4239/5239

Advanced Computer Graphics

Spring 2020

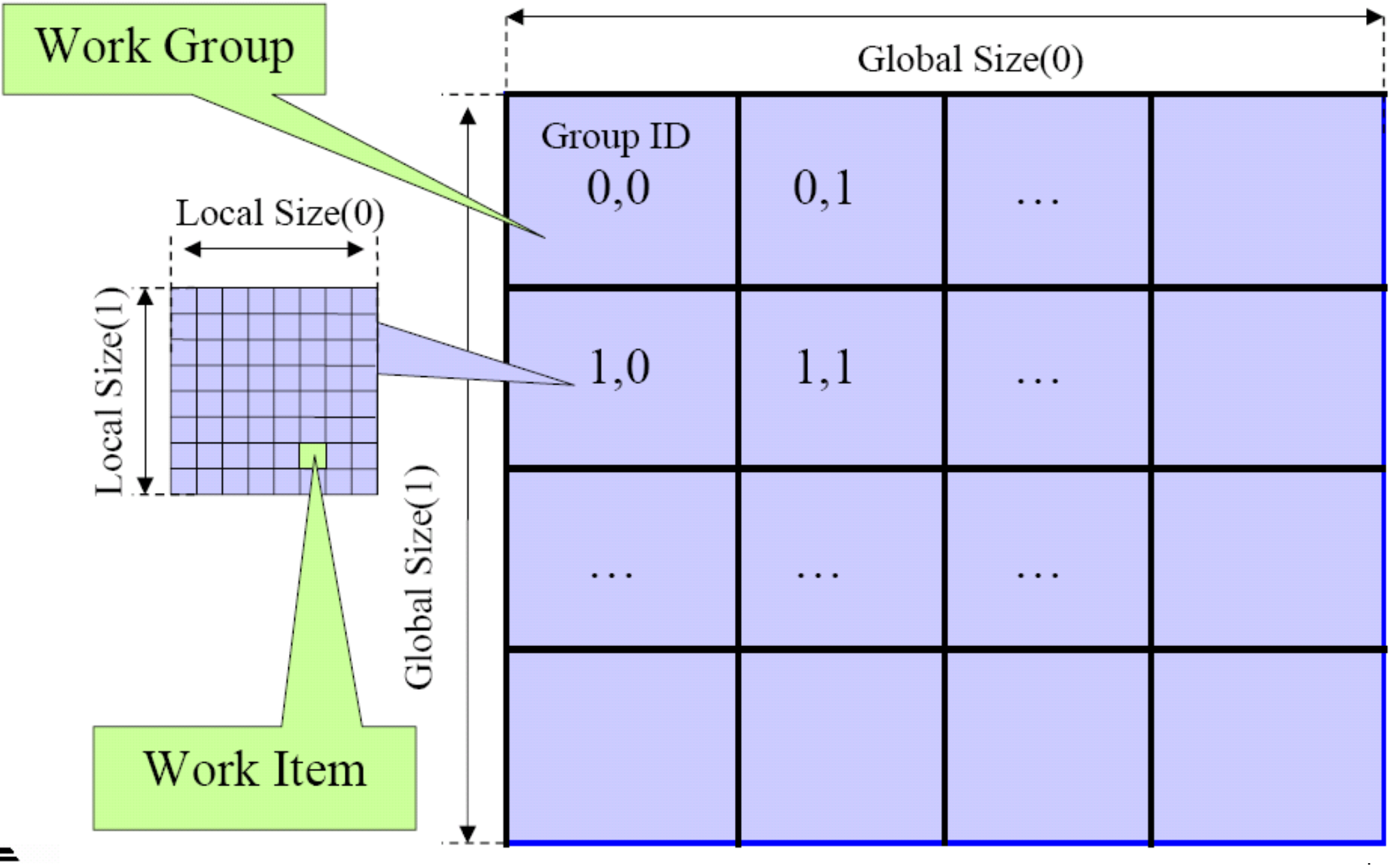
What is OpenCL

- **Open C**omputing **L**anguage
  - Initially developed by Apple
  - Now managed by Khronos
- Parallels CUDA
  - Not just NVIDIA hardware
  - Supports CPU, GPU and Accelerators
  - Conceptually the same, API and syntax different
    - Using OpenCL a bit more tedious than CUDA

OpenCL to CUDA Data Parallelism Model Mapping

OpenCL Parallelism Concept	CUDA Equivalent
kernel	kernel
host program	host program
NDRange (index space)	grid
work item	thread
work group	block

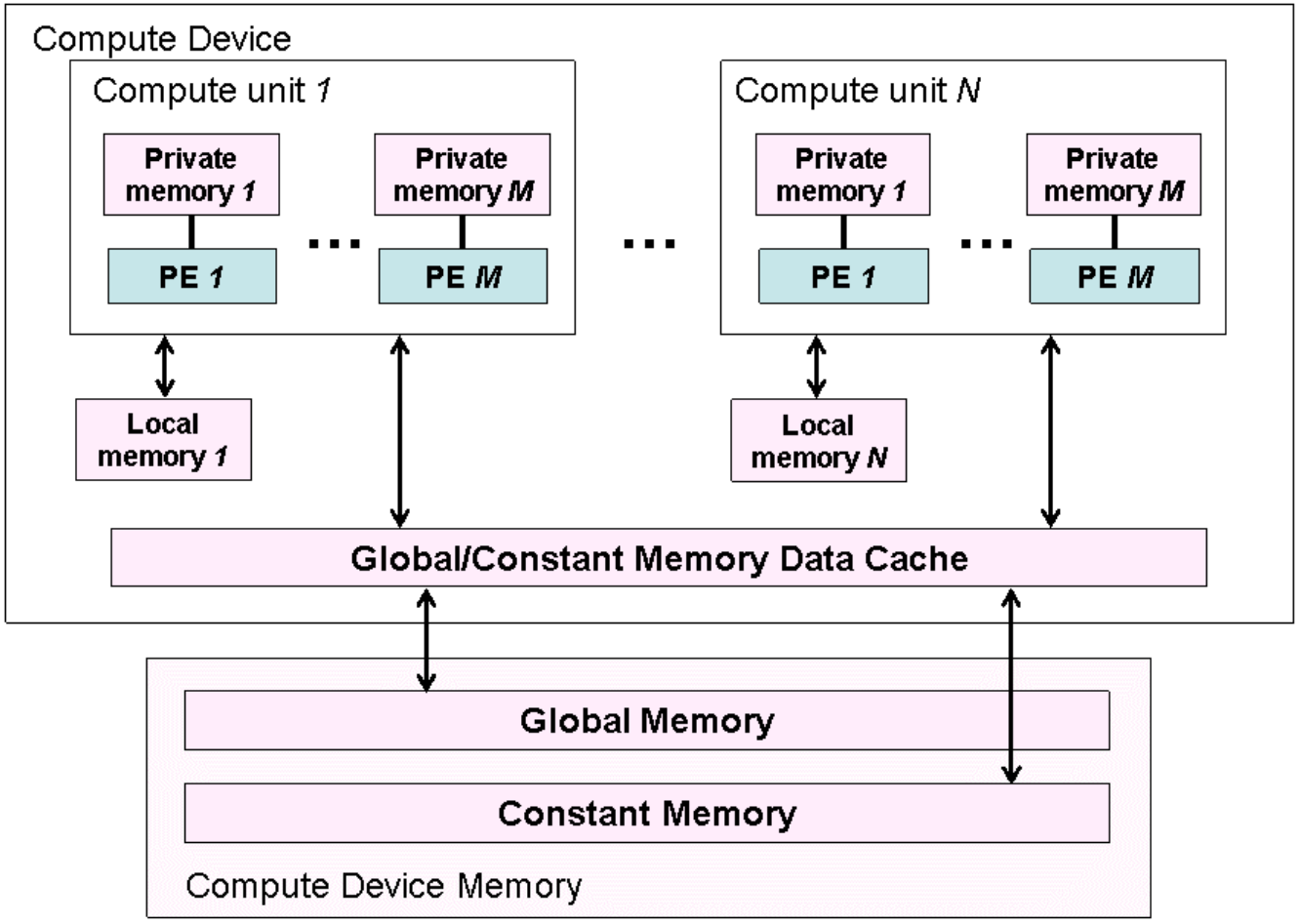
Overview of OpenCL Execution Model



Mapping of OpenCL Dimensions and Indices to CUDA

OpenCL API Call	Explanation	CUDA Equivalent
get_global_id(0);	global index of the work item in the x dimension	blockIdx.x×blockDim.x+threadIdx.x
get_local_id(0)	local index of the work item within the work group in the x dimension	blockIdx.x
get_global_size(0);	size of NDRange in the x dimension	gridDim.x ×blockDim.x
get_local_size(0);	Size of each work group in the x dimension	blockDim.x

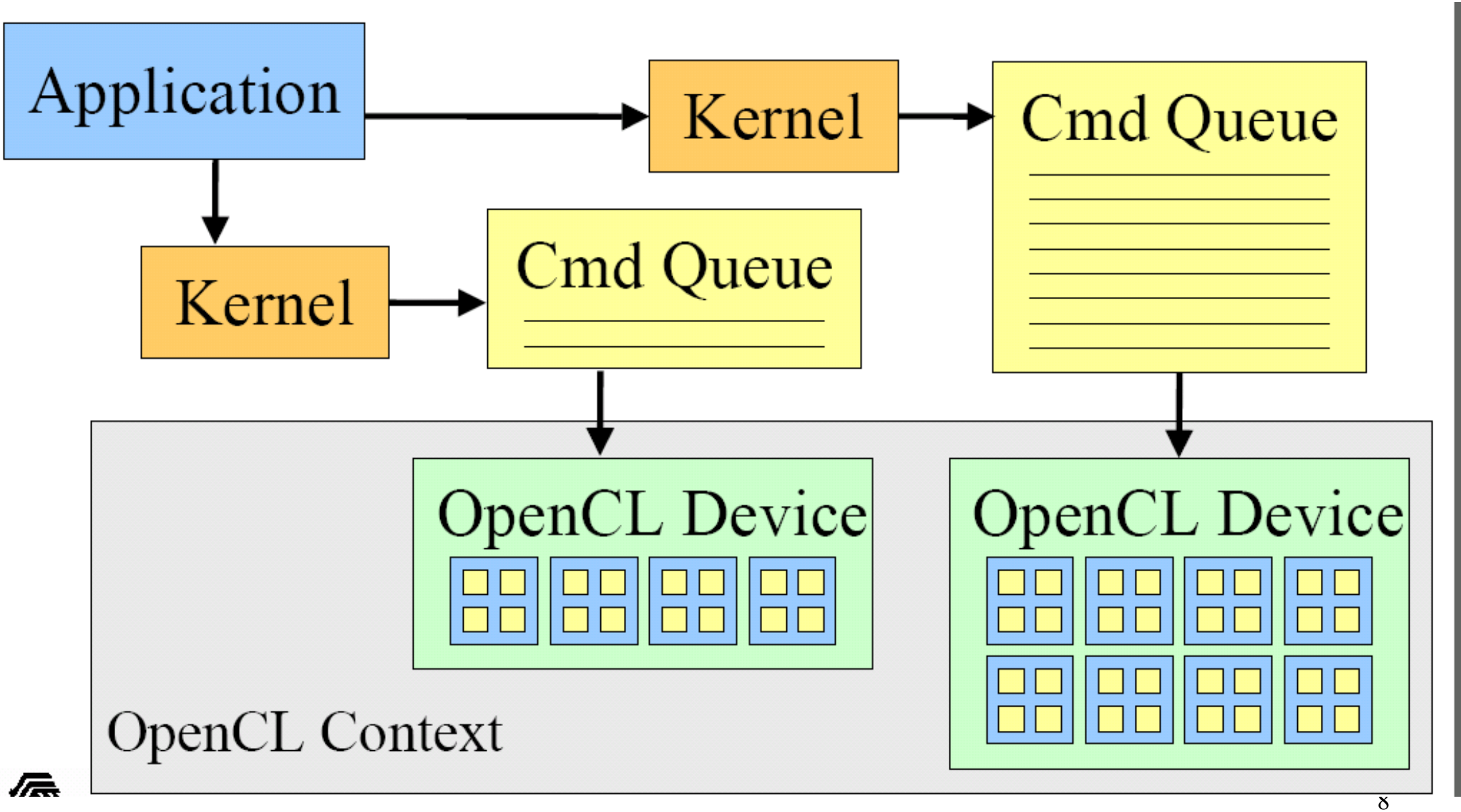
Conceptual OpenCL Device Architecture



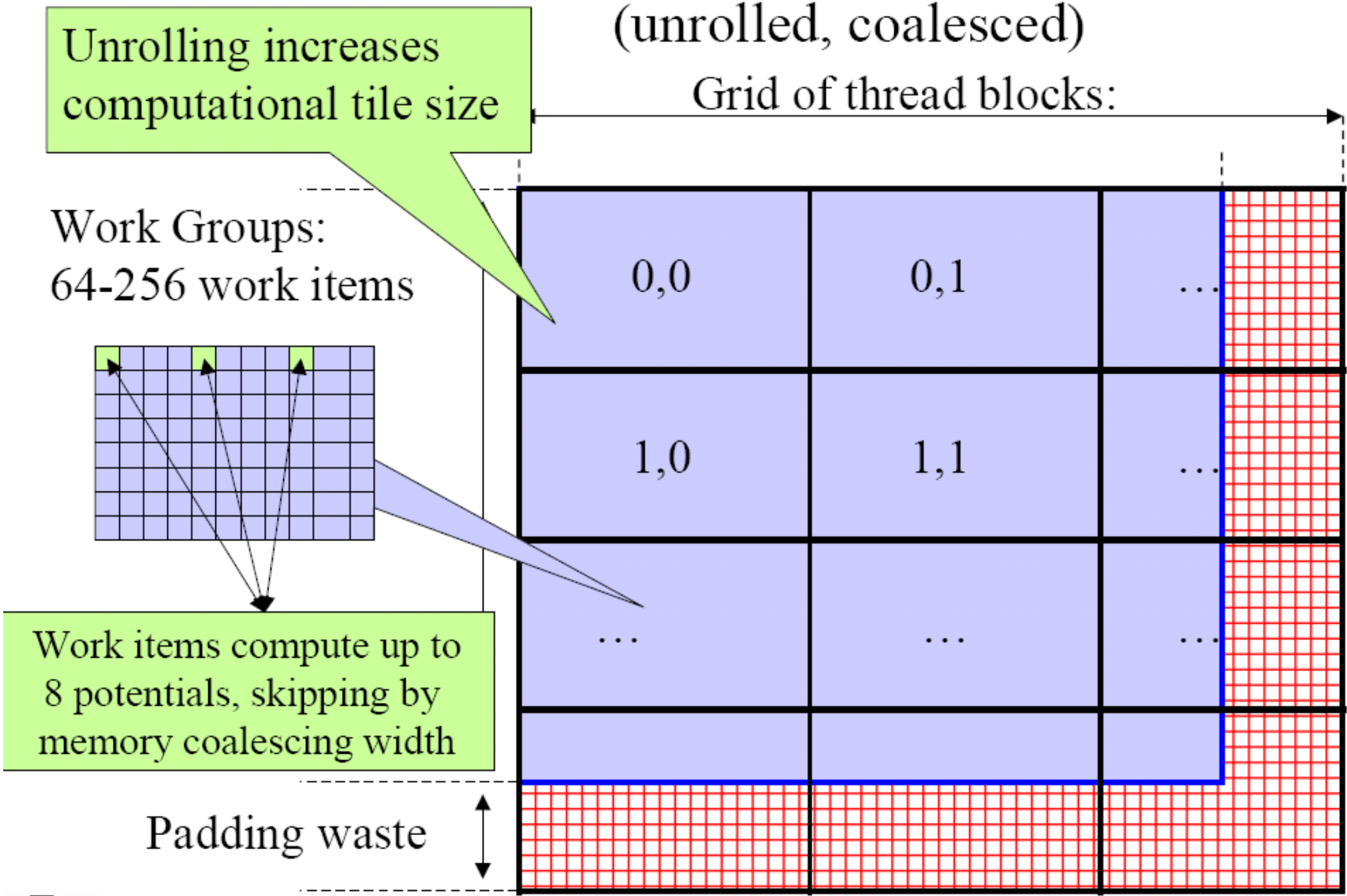
Mapping OpenCL Memory Types to CUDA

OpenCL Memory Types	CUDA Equivalent
global memory	global memory
constant memory	constant memory
local memory	shared memory
private memory	Local memory

OpenCL Context for Device Management



OpenCL Version of DCS Kernel 3



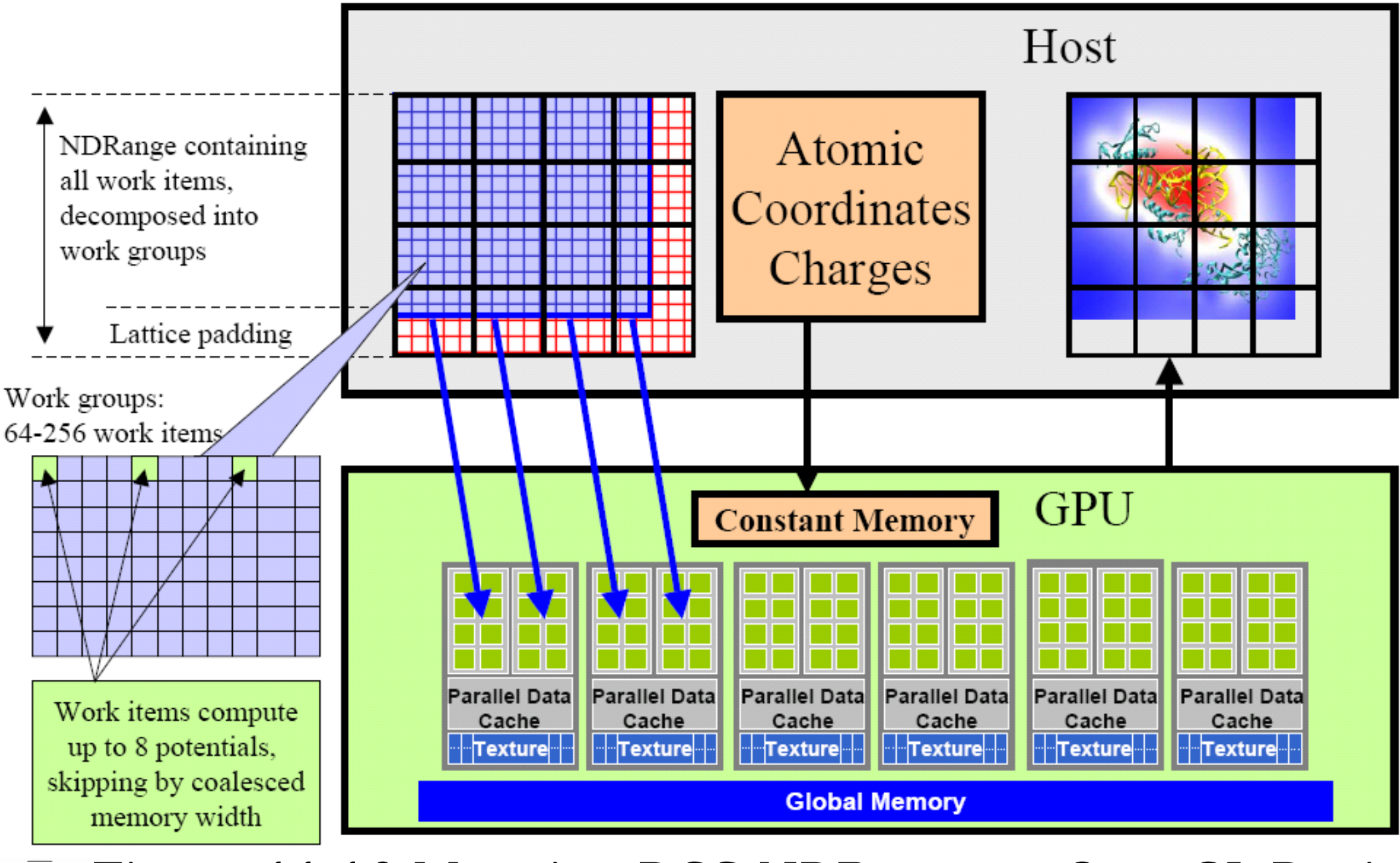


Figure 11.10 Mapping DCS NDRange to OpenCL Devic

Compiling OpenCL Programs

- kernel is compiled at run time
  - Compiler supplied by OpenCL+device driver
    - Easier to support variety of devices
    - Similar to shaders in OpenGL
  - No need for specialized compiler (nvcc)
- Download SDK from Apple/NVIDIA/AMD/...
  - Supports Linux/OSX/Windows
  - All CUDA capable NVIDIA Hardware
  - Recent AMD/ATI hardware
  - Others (e.g. S3 Chrome)

Ex 24: OpenCL Matrix Multiply

- Ex 23 (CUDA Matrix Multiply) ported to OpenCL
- InitGPU: initialization
- AxBd: Multiply AB
  - Copy matrices from host to device
  - Run kernel
  - Copy result back to host
- AxB: kernel
  - Calculate one element (row • column)

Homework 10: GPU Computing

- Make sure that you check that the answer you get is correct
- Just doing meaningless computations on the answer is not acceptable
- To see a speed gain the problem must be big enough