General Purpose computing on GPUs

Paul Springer

Aachen Institute for Advanced Study in
Computational Engineering Science

Aachen, 16.05.13

Outline

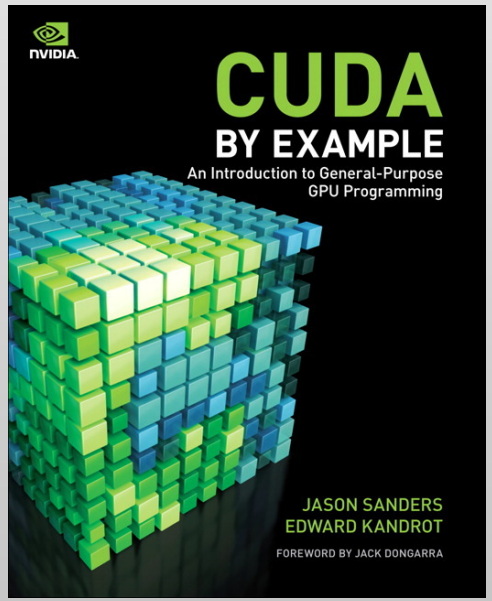
Outline

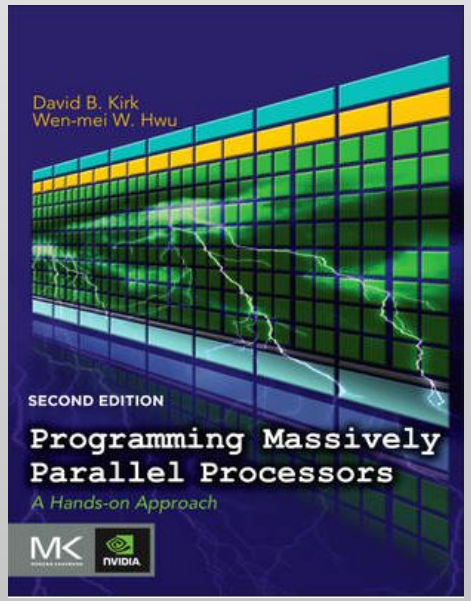
RZ GPU Cluster

- Hardware
  - 27 compute nodes with two NVIDIA Quadro 6000 GPUs (i.e. Fermi)
  - Intel Xeon Westmere EP (X5650) (i.e. 12 cores)
  - Three interactive nodes
- Access
  - Register: servicedesk@rz.rwth-aachen.de
- Usage
  1. ssh <tim-id>@cluster.rz.rwth-aachen.de
  2. ssh linuxgpud[1,2,3]
  3. module load cuda

Resources

- http://docs.nvidia.com/cuda/index.html
  - CUDA C Programming Guide
  - CUDA C Best Practices Guide
- Webinars
  - https://developer.nvidia.com/gpu-computing-webinars
- CUDA SDK samples





Outline

Brief History

- Before 2006
  - Specialized hardware (e.g. computer games)
  - Programmable through OpenGL or Direct3D
- 2006
  - NVIDIA Compute Unified Device Architecture (CUDA)
  - NVIDIA Tesla Architecture
- 2008
  - Khronos Group's Open Computing Language (OpenCL)
    - AMD, IBM, Intel, NVIDIA and Apple
- 2009
  - NVIDIA Fermi Architecture
    - Higher DP performance
- 2012
  - NVIDIA Kepler Architecture

Frequency wall + Memory wall + Energy wall

=

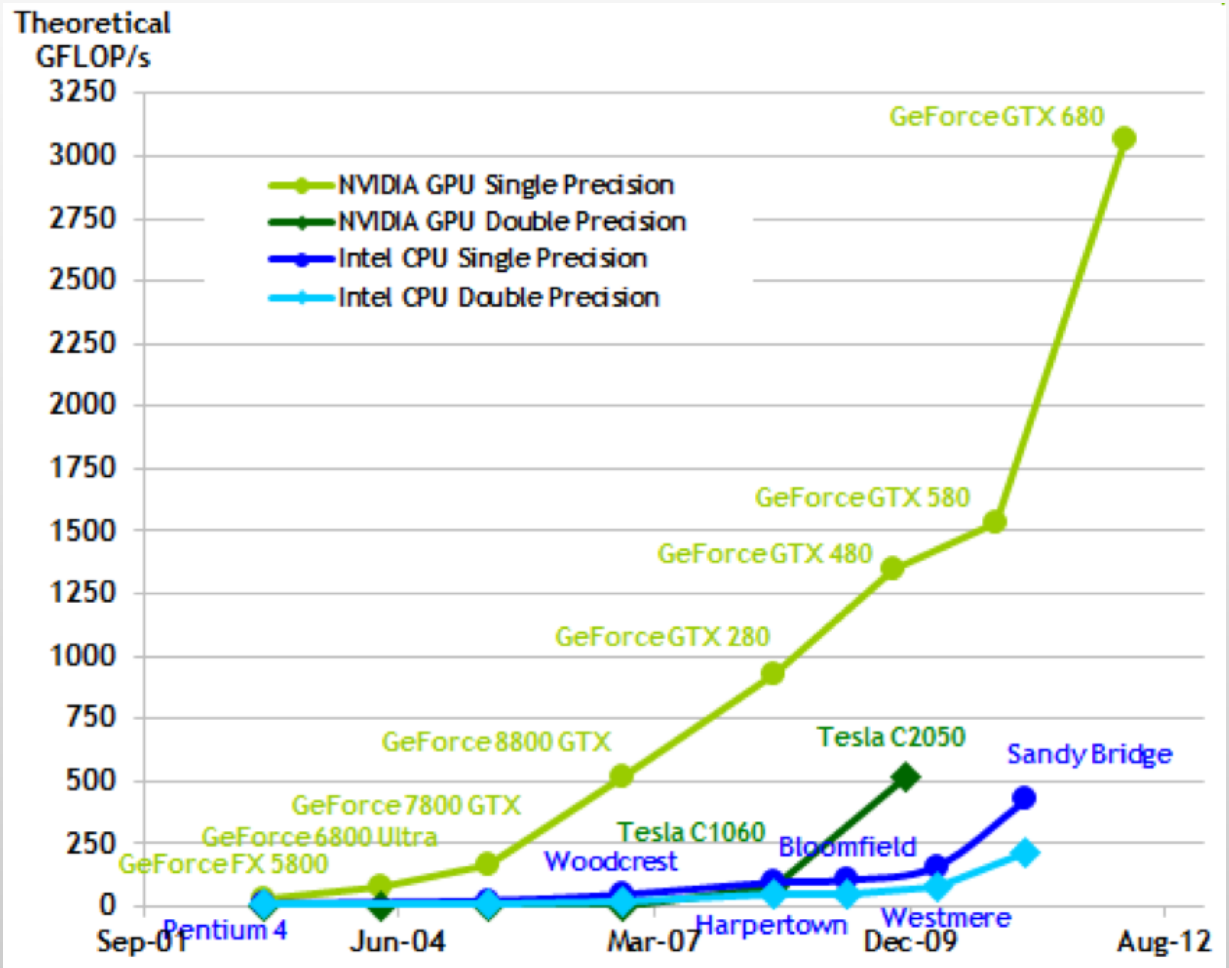


Figure: Theoretical peak SP performance. Taken from [3].

Why GPUs?

- Speedup applications
- Solve bigger problem sizes
- Solve problems with real-time constraints
- Top 500 List
  - Contains 62 heterogeneous systems
  - Titan @ Oak Ridge National Lab (17.6 PFLOPS/s)
    - NVIDIA K20X GPU accelerator
  - Stampede @ Texas Advanced Computing Center (2.7 PFLOPS/s)
    - Intel Xeon Phi co-processor

©Oak Ridge National Laboratory

- High power efficiency

Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)
1	2,499.44	National Institute for Computational Sciences/University of Tennessee	Beacon - Appro GreenBlade GB824M, Xeon E5-2670 8C 2.600GHz, Infiniband FDR, Intel Xeon Phi 5110P	44.89
2	2,351.10	King Abdulaziz City for Science and Technology	SANAM - Adtech ESC4000/FDR G2, Xeon E5-2650 8C 2.000GHz, Infiniband FDR, AMD FirePro S10000	179.15
3	2,142.77	DOE/SC/Oak Ridge National Laboratory	Titan - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x	8,209.00
4	2,121.71	Swiss Scientific Computing Center (CSCS)	Todi - Cray XK7 , Opteron 6272 16C 2.100GHz, Cray Gemini interconnect, NVIDIA Tesla K20 Kepler	129.00
5	2,102.12	Forschungszentrum Juelich (FZJ)	JUQUEEN - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect	1,970.00
6	2,101.39	Southern Ontario Smart Computing Innovation Consortium/University of Toronto	BGQdev - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect	41.09
7	2,101.39	DOE/NNSA/LLNL	rzuseq - BlueGene/Q, Power BQC 16C 1.60GHz, Custom	41.09
8	2,101.39	IBM Thomas J. Watson Research Center	BlueGene/Q, Power BQC 16C 1.60GHz, Custom	41.09
9	2,101.12	IBM Thomas J. Watson Research Center	BlueGene/Q, Power BQC 16C 1.60 GHz, Custom	82.19
10	2,101.12	Ecole Polytechnique Federale de Lausanne	CADMOS BG/Q - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect	82.19

Figure: Top 10 of Green500 November 2012.

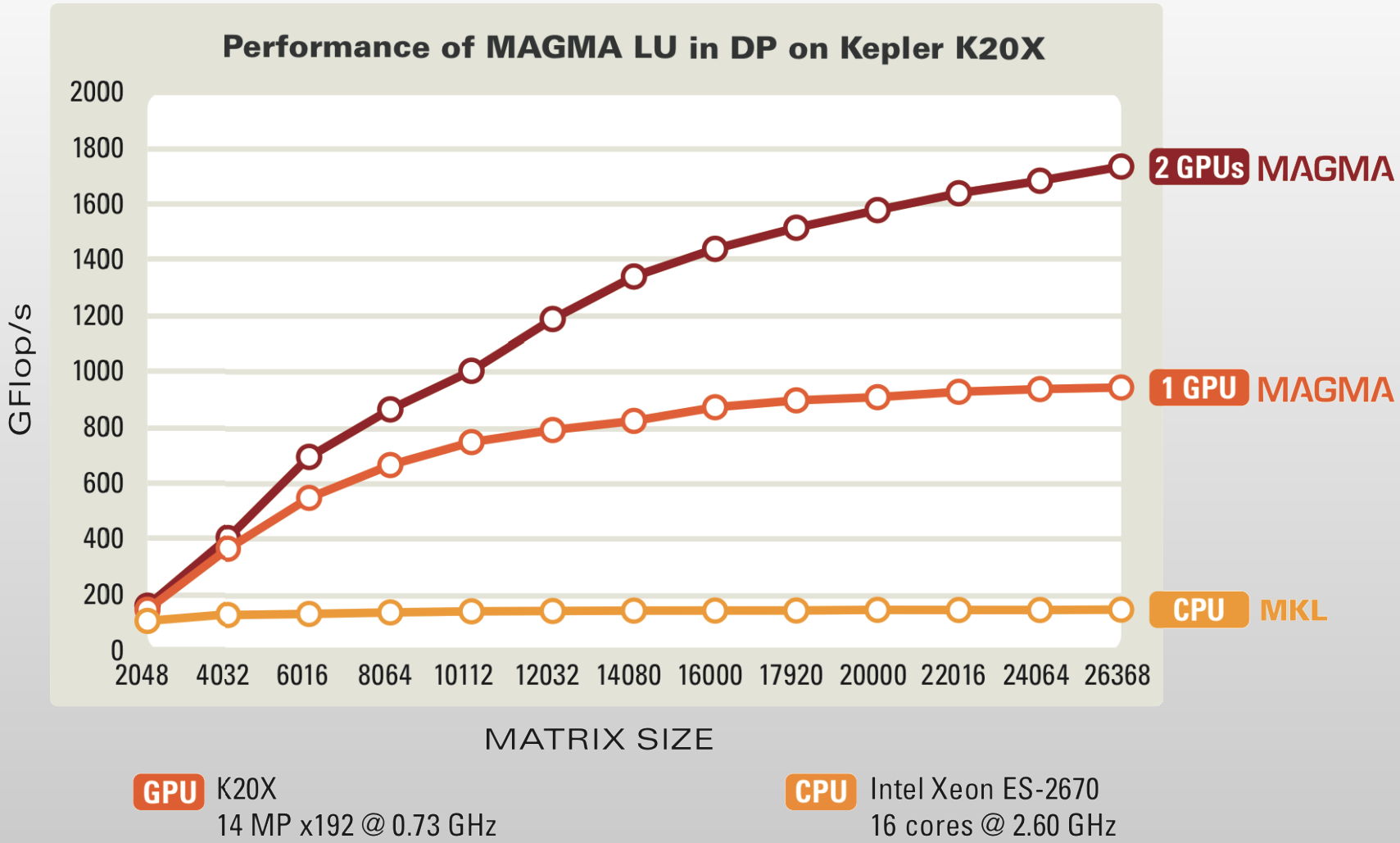
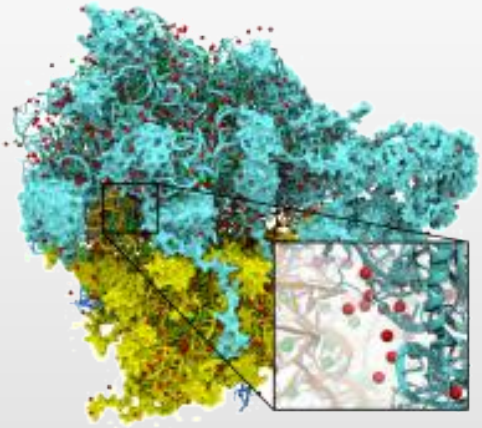
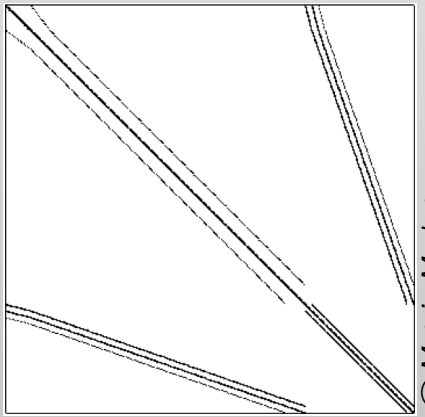


Figure: MAGMA LU Performance [1].

- Dense/ Sparse Linear Algebra
- Molecular Dynamics Simulations
- Monte Carlo Simulations
- Medical imaging (e.g. MRI)
- Fluid dynamics
- Cryptographic hash functions
- Finite Elements
- Iterative Solvers
- ...



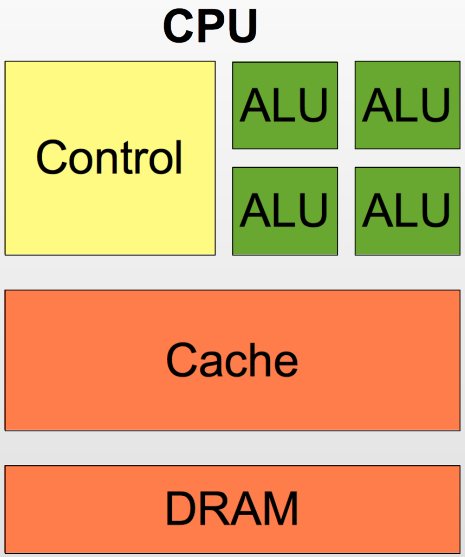


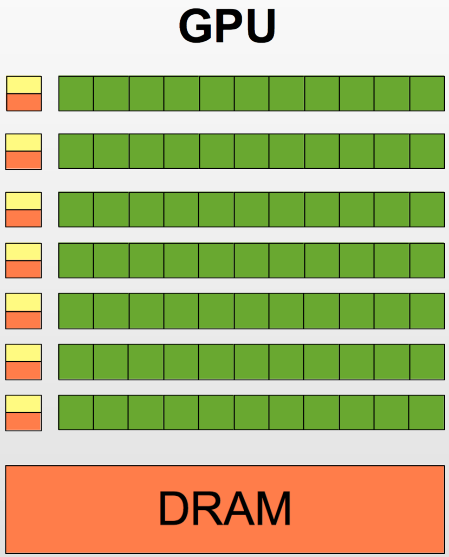
© Matrix Market

Outline

CPUs vs GPUs

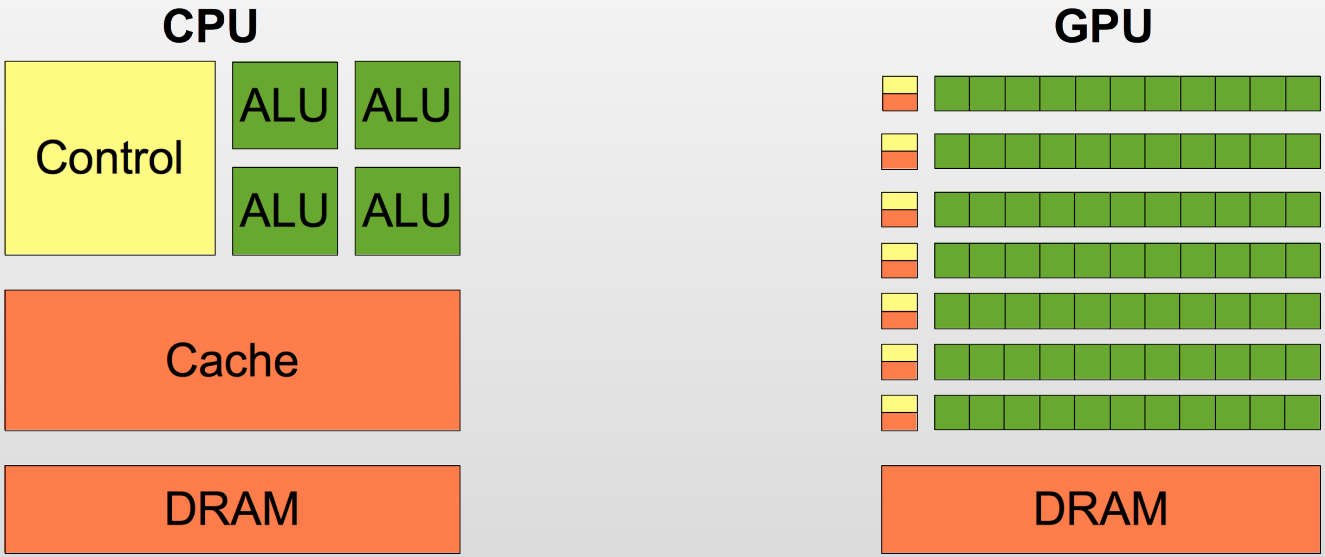
CPUs vs GPUs



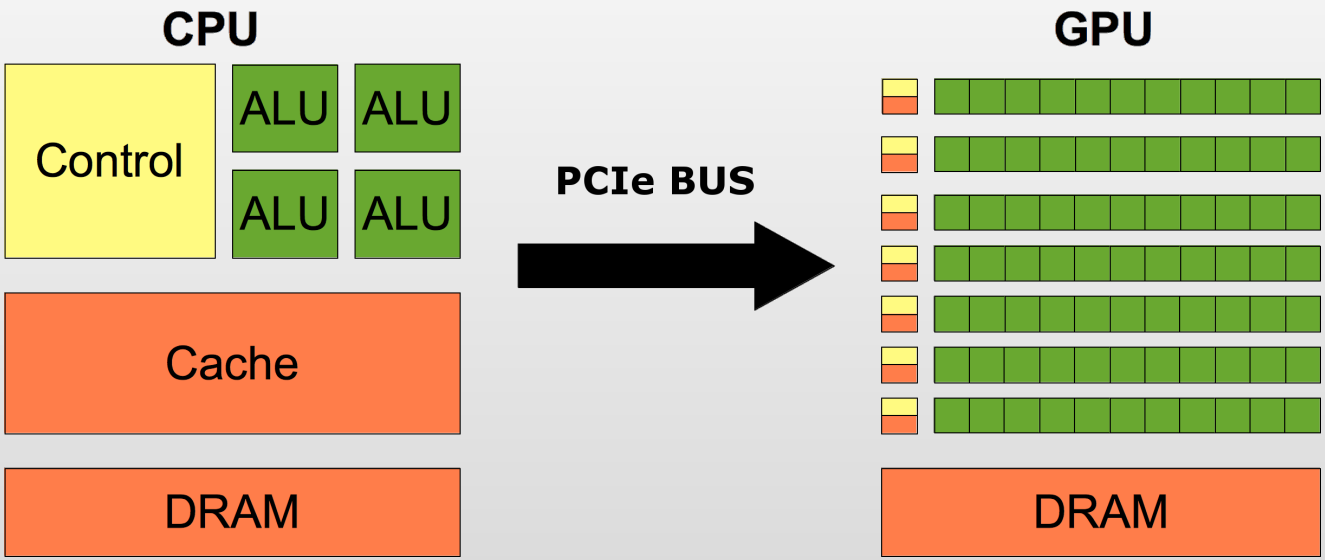


- $\approx <$ 20 cores @ high f
- Few threads
- Rich cache hierarchy
- Many control units
- 8 SP vector-width
- Large amount of memory per thread

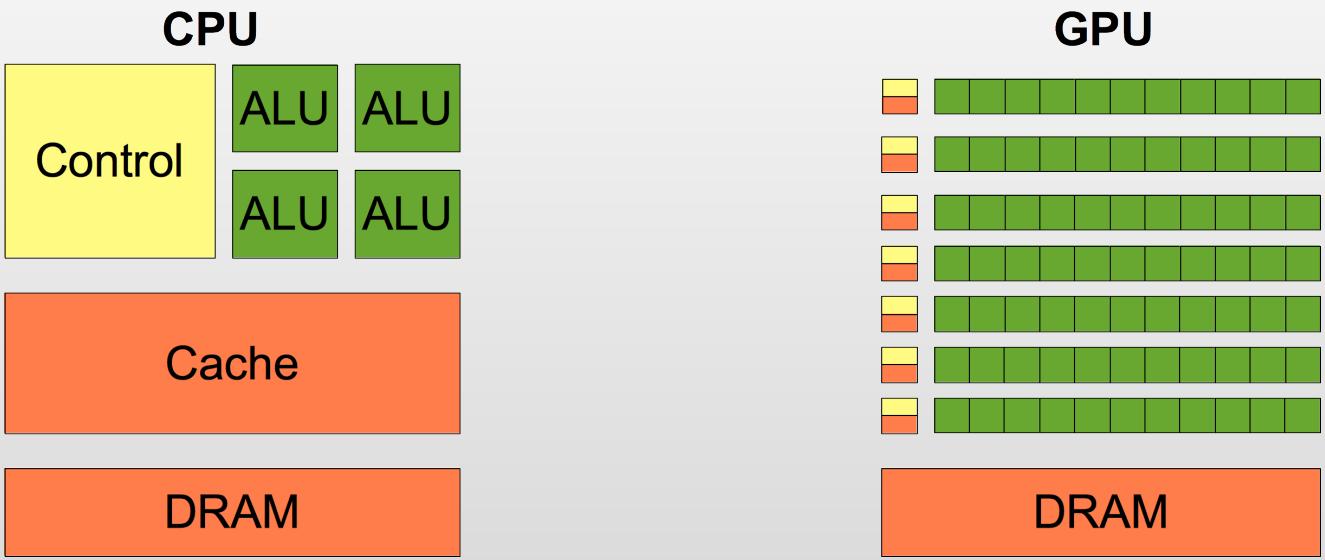
- $\approx >$ 1000 cores @ low f
- Thousands of threads
- Small caches
- Few control units
- 32 SP vector-width



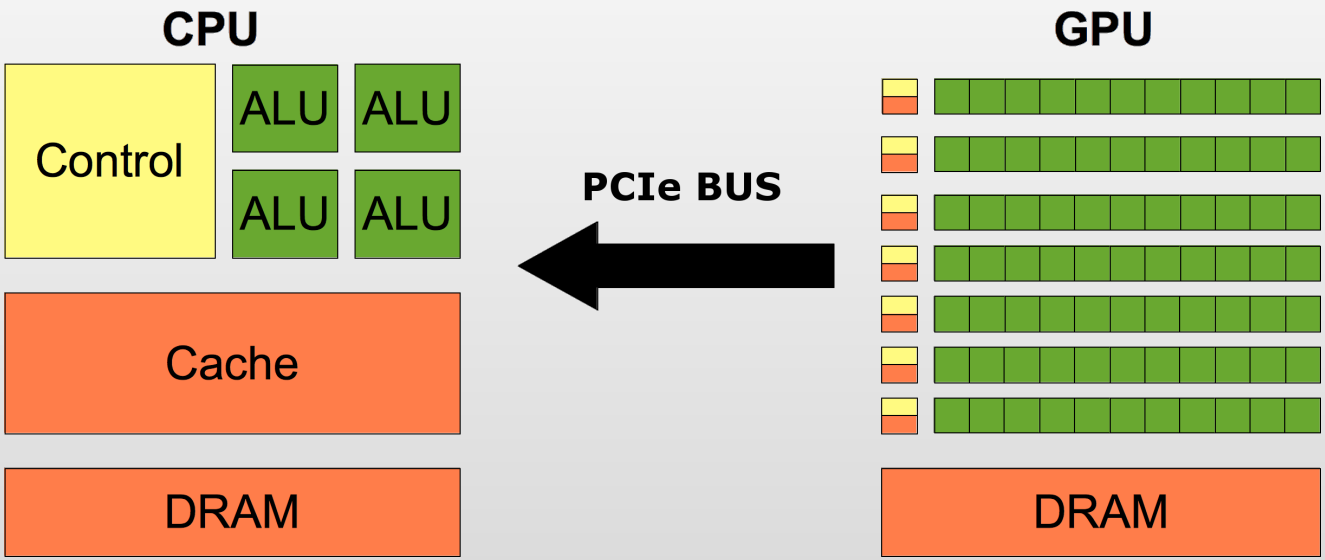
- Separate physical memory space
- Asynchronous execution
  - Possible to overlap CPU and GPU computations



1. Transfer data from *host* to *device*



1. Transfer data from *host* to *device*
2. Process data



1. Transfer data from *host* to *device*
2. Process data
3. Transfer data from *device* to *host*

```
Running on...

 Device 0: Quadro 6000
 Quick Mode

 Host to Device Bandwidth, 1 Device(s)
 PINNED Memory Transfers
   Transfer Size (Bytes)        Bandwidth(MB/s)
   33554432                     5742.4

 Device to Host Bandwidth, 1 Device(s)
 PINNED Memory Transfers
   Transfer Size (Bytes)        Bandwidth(MB/s)
   33554432                     6225.0

 Device to Device Bandwidth, 1 Device(s)
 PINNED Memory Transfers
   Transfer Size (Bytes)        Bandwidth(MB/s)
   33554432                     97456.9
```

Figure: CUDA SDK bandwidthTest @ NVIDIA Quadro 6000 GPU.

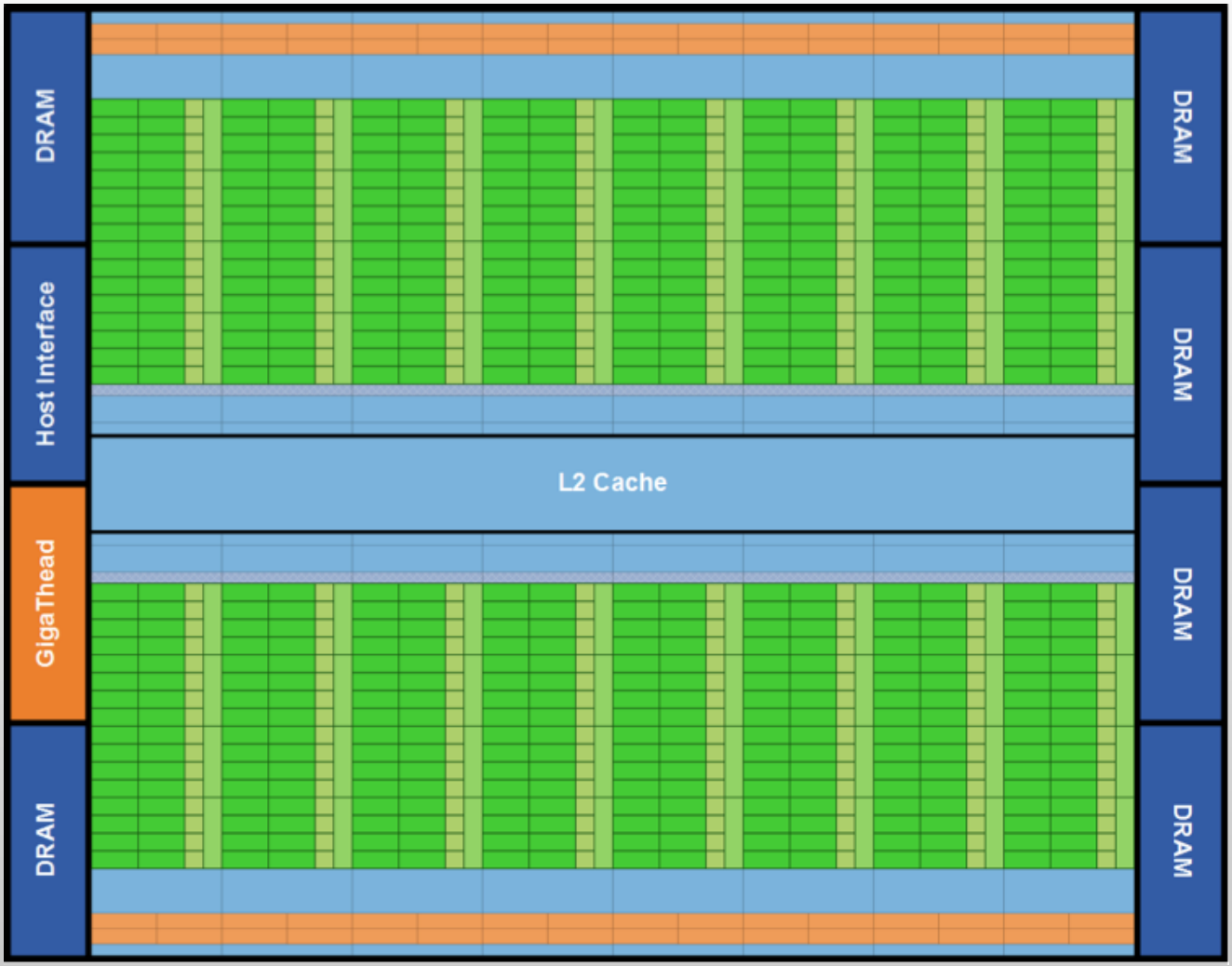
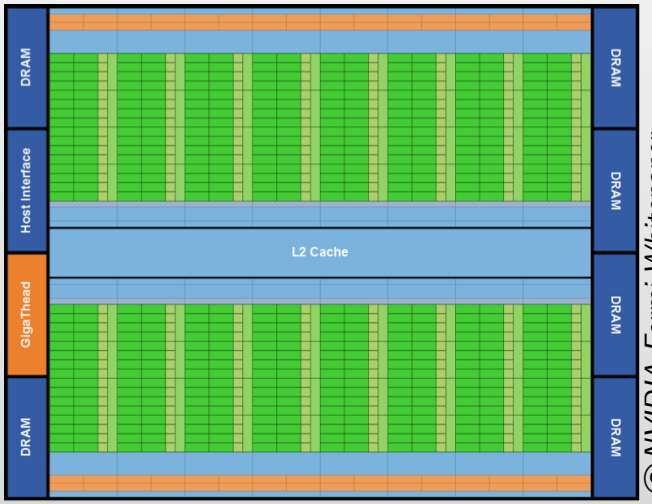
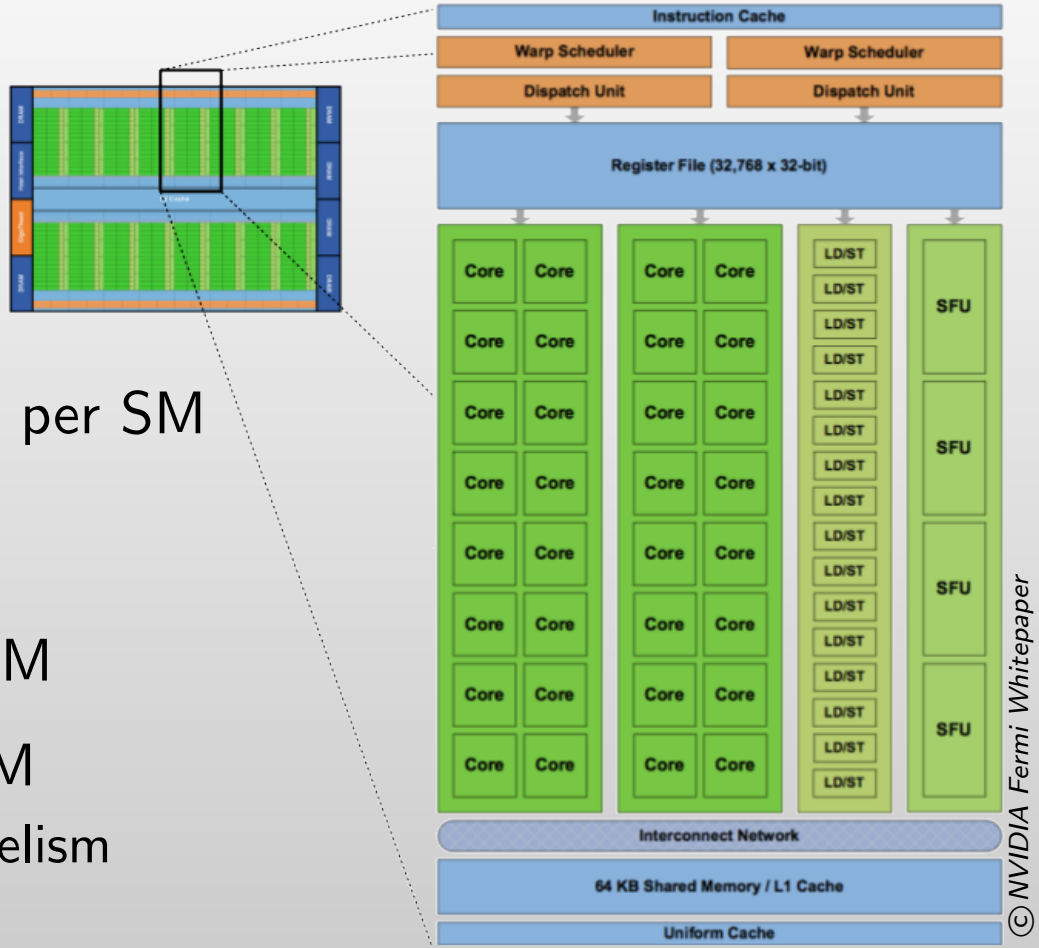


Figure: NVIDIA Fermi - Overview [2].

- Compute capability 2.x
- 14-16 SMs
- 448-512 cores @ 1.15 Ghz
- Up to 1331 GFLOPS SP
- Up to 665 GFLOPS DP
- Up to 6 GB global memory
  - Latency: 300-600 cycles
  - Bandwidth 148 GB/sec
- True cache hierarchy
- Error-correcting code (ECC) support



© NVIDIA Fermi Whitepaper



- 48/16 KB *shared memory* per SM
  - Low latency
  - High throughput
- 32K 4-byte registers per SM
- Up to 1536 threads per SM
  - High thread-level parallelism
  - **Hide memory latency**

```
Device 0: "Quadro 6000"
  CUDA Driver Version / Runtime Version          5.0 / 5.0
  CUDA Capability Major/Minor version number:    2.0
  Total amount of global memory:                 5375 MBytes (5636554752 bytes)
  (14) Multiprocessors x ( 32) CUDA Cores/MP:    448 CUDA Cores
  GPU Clock rate:                                1147 MHz (1.15 GHz)
  Memory Clock rate:                             1494 Mhz
  Memory Bus Width:                              384-bit
  L2 Cache Size:                                 786432 bytes
  Max Texture Dimension Size (x,y,z)             1D=(65536), 2D=(65536,65535), 3D=(2048,2048,2048)
  Max Layered Texture Size (dim) x layers        1D=(16384) x 2048, 2D=(16384,16384) x 2048
  Total amount of constant memory:               65536 bytes
  Total amount of shared memory per block:       49152 bytes
  Total number of registers available per block: 32768
  Warp size:                                     32
  Maximum number of threads per multiprocessor:  1536
  Maximum number of threads per block:           1024
  Maximum sizes of each dimension of a block:    1024 x 1024 x 64
  Maximum sizes of each dimension of a grid:     65535 x 65535 x 65535
  Maximum memory pitch:                          2147483647 bytes
  Texture alignment:                             512 bytes
  Concurrent copy and kernel execution:          Yes with 2 copy engine(s)
  Run time limit on kernels:                     Yes
  Integrated GPU sharing Host Memory:            No
  Support host page-locked memory mapping:       Yes
  Alignment requirement for Surfaces:            Yes
  Device has ECC support:                        Enabled
  Device supports Unified Addressing (UVA):      Yes
  Device PCI Bus ID / PCI location ID:           2 / 0
  Compute Mode:
     < Exclusive Process (many threads in one process is able to use ::cudaSetDevice() with this device) >
```

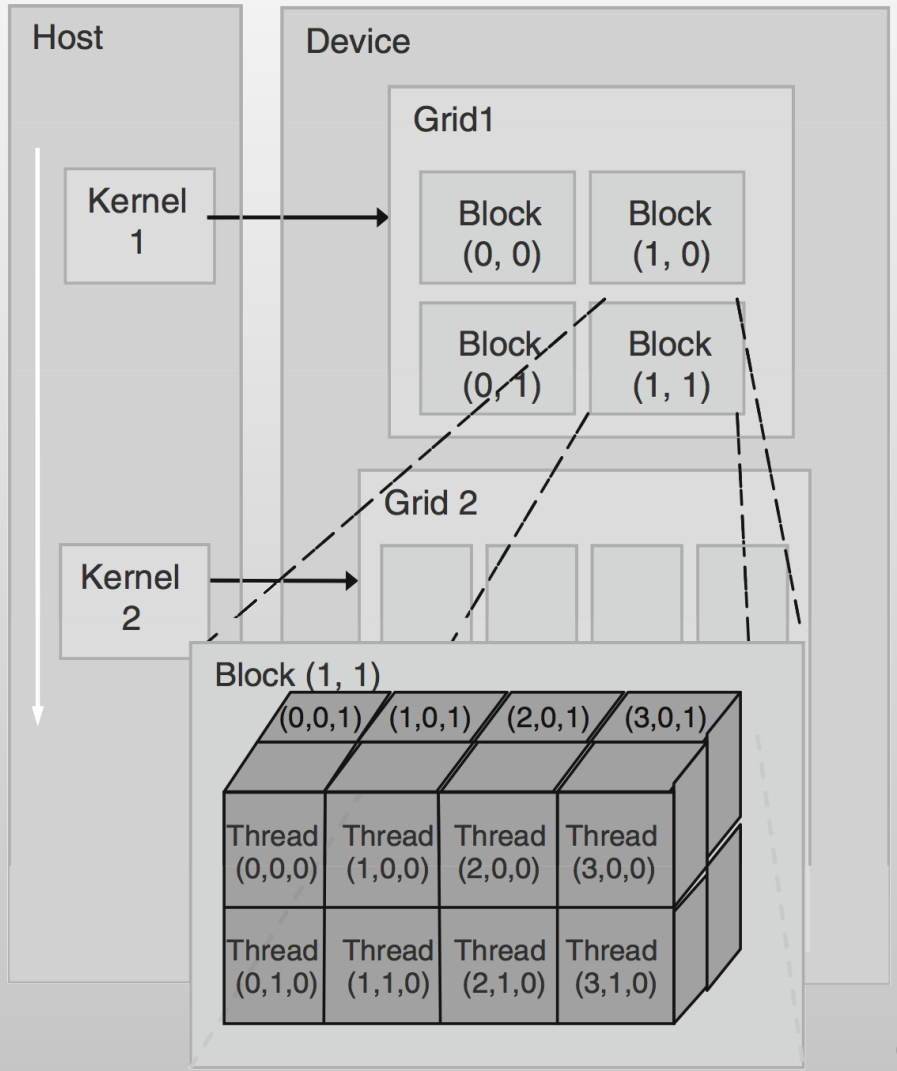
Figure: CUDA SDK deviceQuery @ NVIDIA Quadro 6000 GPU.

Outline

- C/C++ API
- Only for NVIDIA GPUs
- Interfaces for other languages
  - Python (e.g. pyCUDA)
  - Fortran (via PGI compiler)
- Alternatives
  - OpenCL
  - OpenACC

CUDA Thread Organization

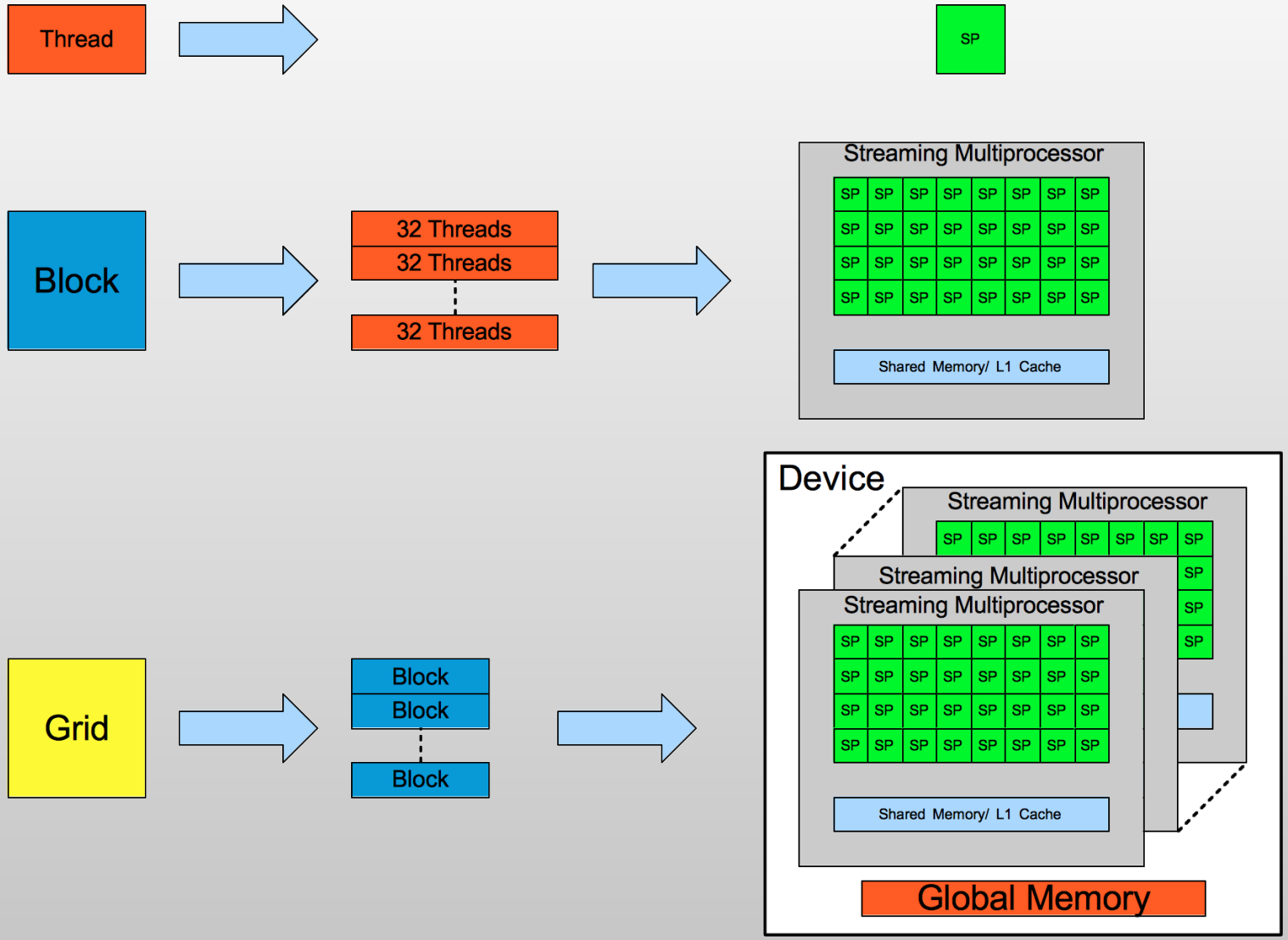
- Execution Units on CPU:
  - Processes
  - Threads
- Execution Units on GPU:
  - Grids
  - Threadblocks
  - Threads



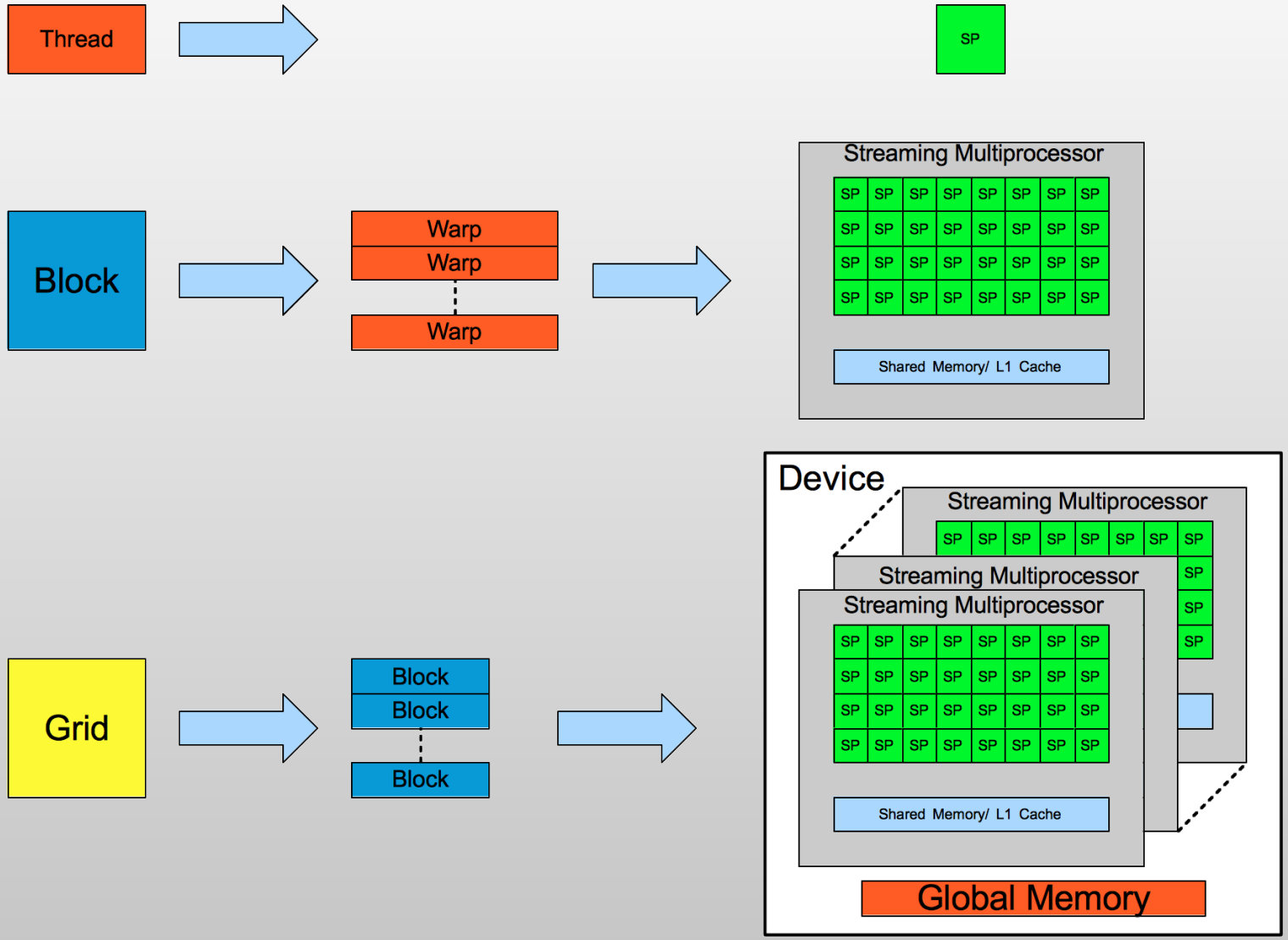
© Programming Massively Parallel Processors

Thread Organization example

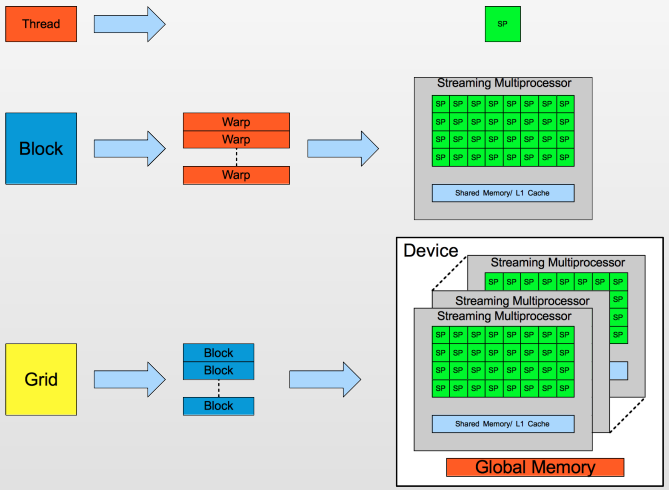
CUDA Execution Model



CUDA Execution Model

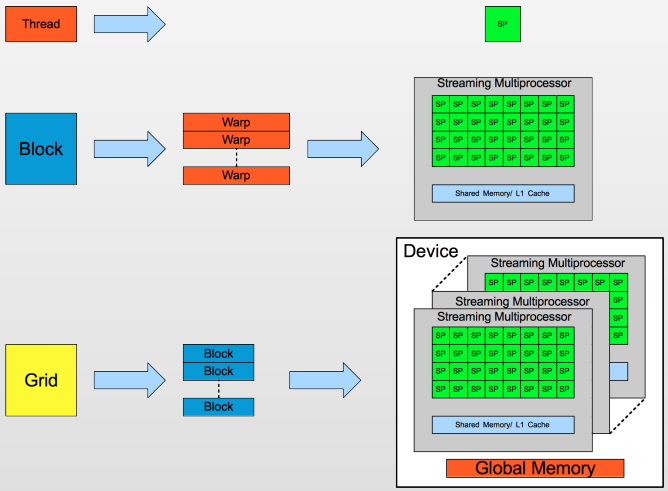


CUDA Execution Model



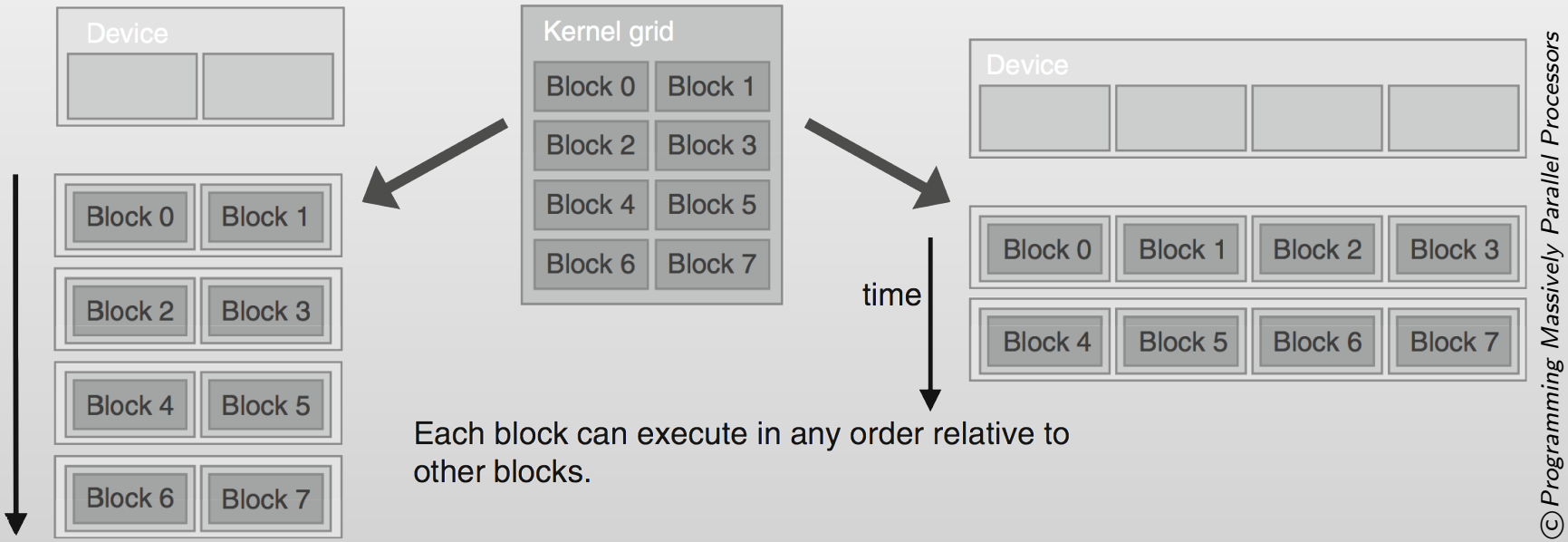
- 32 Threads = 1 *warp*
  - Smallest execution unit
  - Execute in lock-step
- Single Instruction Multiple Thread (SIMT)
  - Similar to SIMD
- Threads within the same threadblock can synchronize

CUDA Execution Model



- Threadblocks can not synchronize!
- Multiple threadblocks run concurrently on the same SM
  - High thread-level parallelism
  - Hide memory latencies
- Multiple grids can be executed simultaneously for CC $\geq$ 2.0

Transparent Scaling



Each block can execute in any order relative to other blocks.

SAXPY example

Outline

References I

[1] MAGMA.
Magma SC2012 Handout.
http://icl.cs.utk.edu/graphics/posters/files/SC12-MAGMA.pdf, May 2013.

[2] NVIDIA.
NVIDIA's Next Generation CUDA Compute Architecture: Fermi.
NVIDIA Whitepaper, 2009.

[3] NVIDIA.
CUDA C Best Practices Guide, October 2012.
Version 5.0.