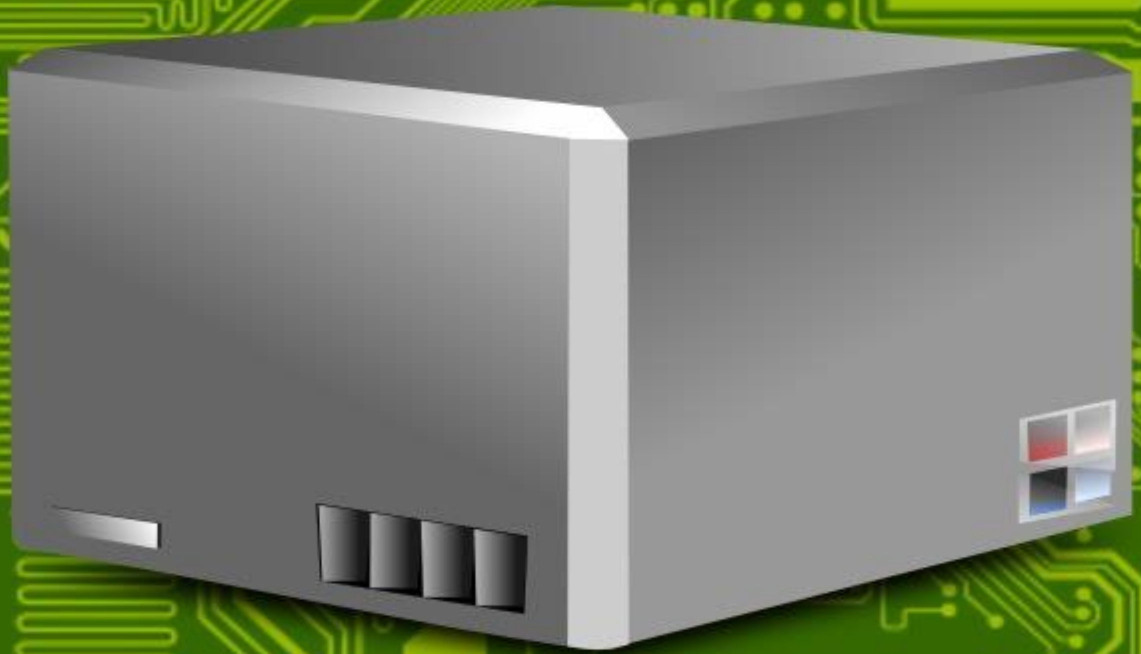
GPU Computing



CUDA 程序优化

Hui Liu
Email: hui.sc.liu@gmail.com

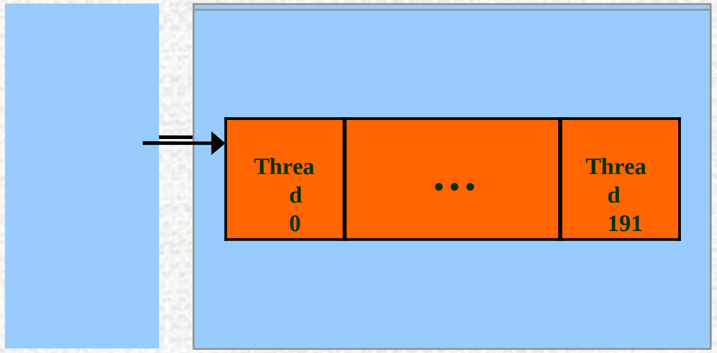
CUDA: 高性能计算

- CUDA 简化 NVIDIA GPUs 大规模并行
  - Direct execution of data-parallel programs
  - Without the overhead of a graphics API

- 理解与优化 CUDA 程序, 获得更高的加速与计算性能
  - 这个视频的目的

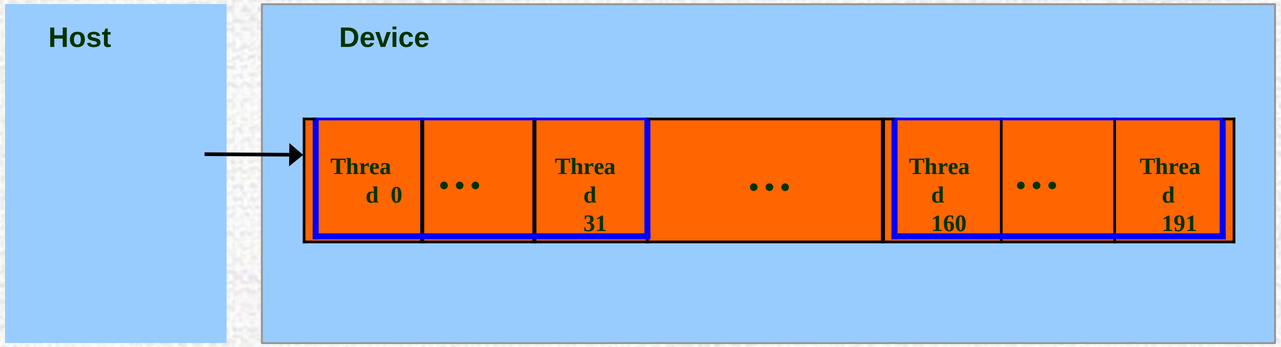
技术术语: 线程, Device Thread

- The GPU (aka *CUDA device*) operates as a coprocessor to the CPU (aka *host*)
- *Thread*: concurrent code and associated state executed on the CUDA device (in parallel with other threads)
  - The unit of parallelism in CUDA
  - Note difference from CPU threads: creation cost, resource usage, and switching cost of GPU threads is much smaller



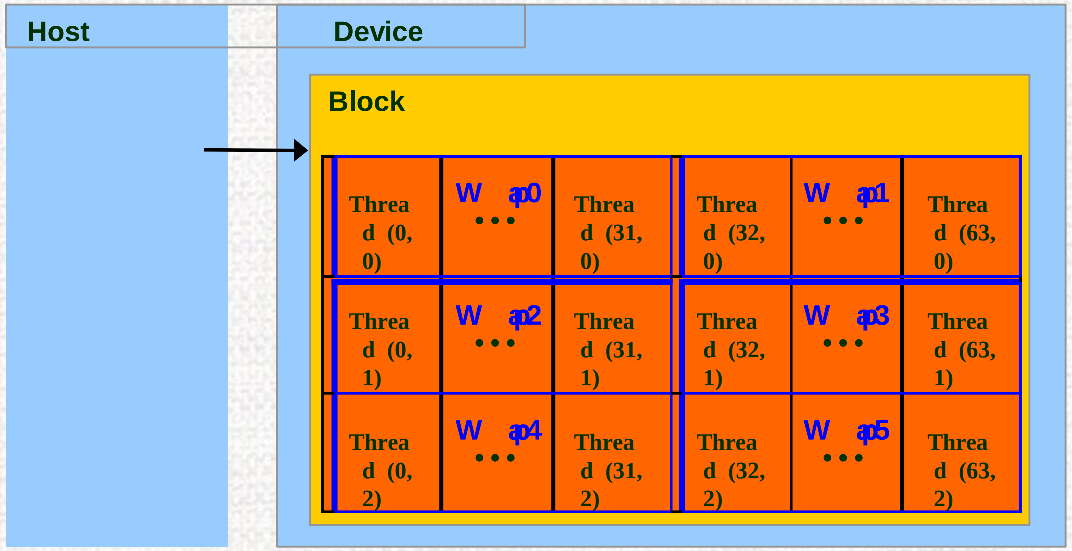
技术术语: 线程束, Warp

- *Warp*: a group of threads executed *physically* in parallel (Single Instruction Multiple Data)
  - Warps are executed *logically* in parallel (execution order undefined)
  - *Half-warp*: the first or second half of a warp
  - *Warp size*: # of threads in a warp (32 threads on G80)



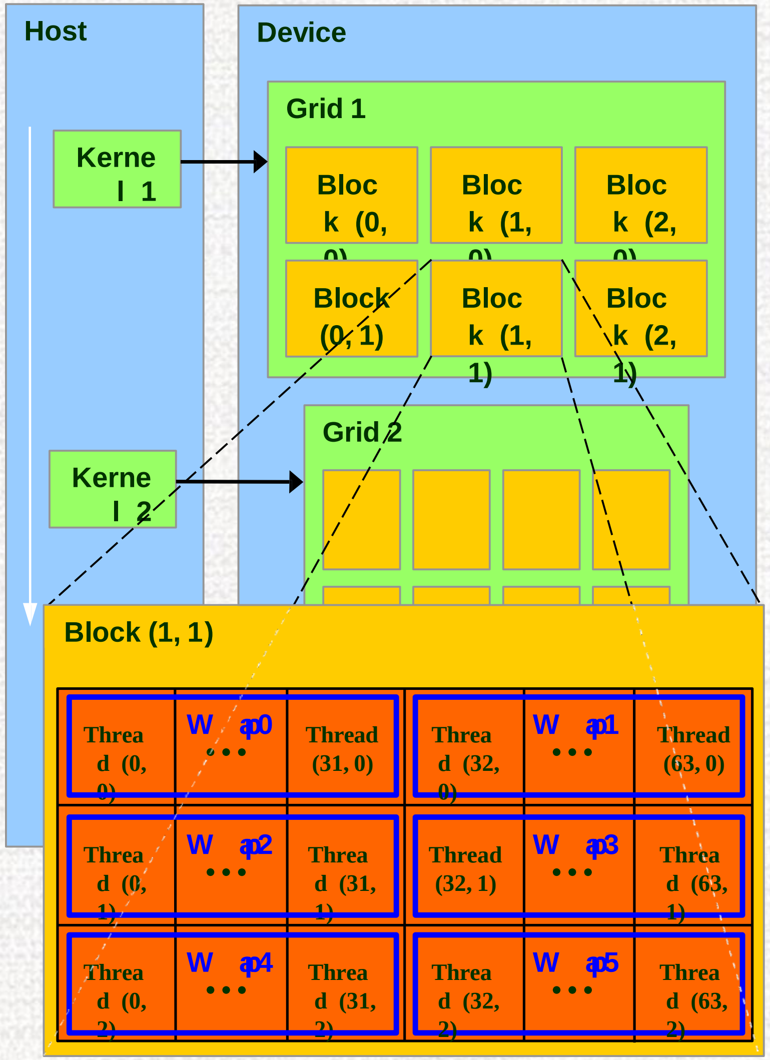
技术术语: 线程块, Thread Block

- *Thread block*: a group of warps that are executed together and can share memory on a single multiprocessor



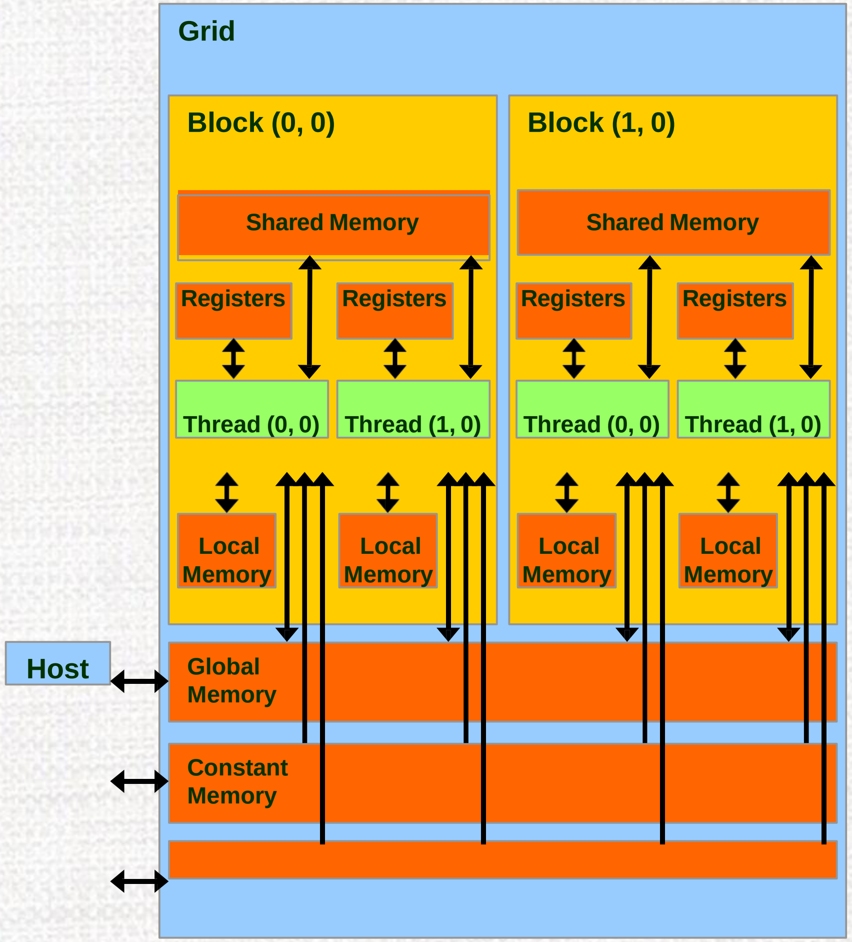
技术术语: 网格与核函数, Grid and Kernel

- *Grid*: a group of thread blocks that execute a single CUDA program (aka *kernel*) *logically* in parallel (Single Program Multiple Data)



内存架构: Memory Architecture

- Host memory
  - Device ↔ host memory bandwidth is 4 GB/s peak (PCI-express x16)
- Global/local device memory
  - High latency, not cached
  - 80 GB/s peak, 1.5 GB (Quadro FX 5600)
- Shared memory
  - On-chip, low latency, very high bandwidth, 16 KB
  - Like a user-managed per- multiprocessor cache
- Texture memory
  - Read-only, high latency, cached
- Constant memory
  - Read-only, low latency, cached, 64 KB



性能优化策略

- 最大化并行执行

- 优化内存使用

- 优化指令

https://github.com/huiscliu/tutorials/CUDA编程入门

THANK YOU