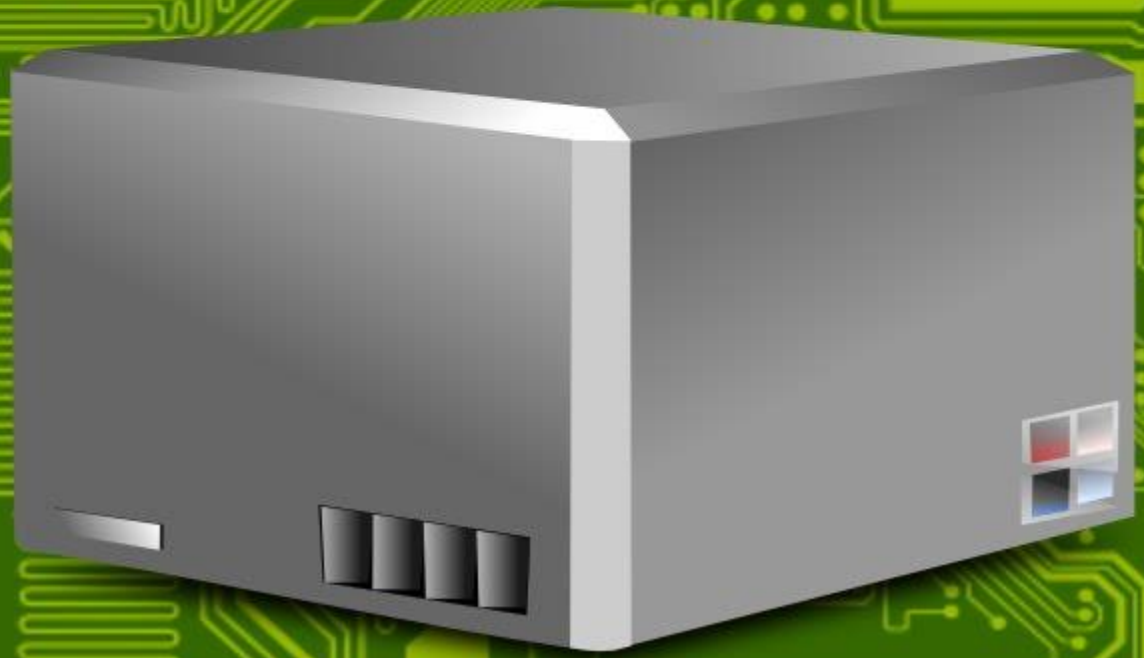
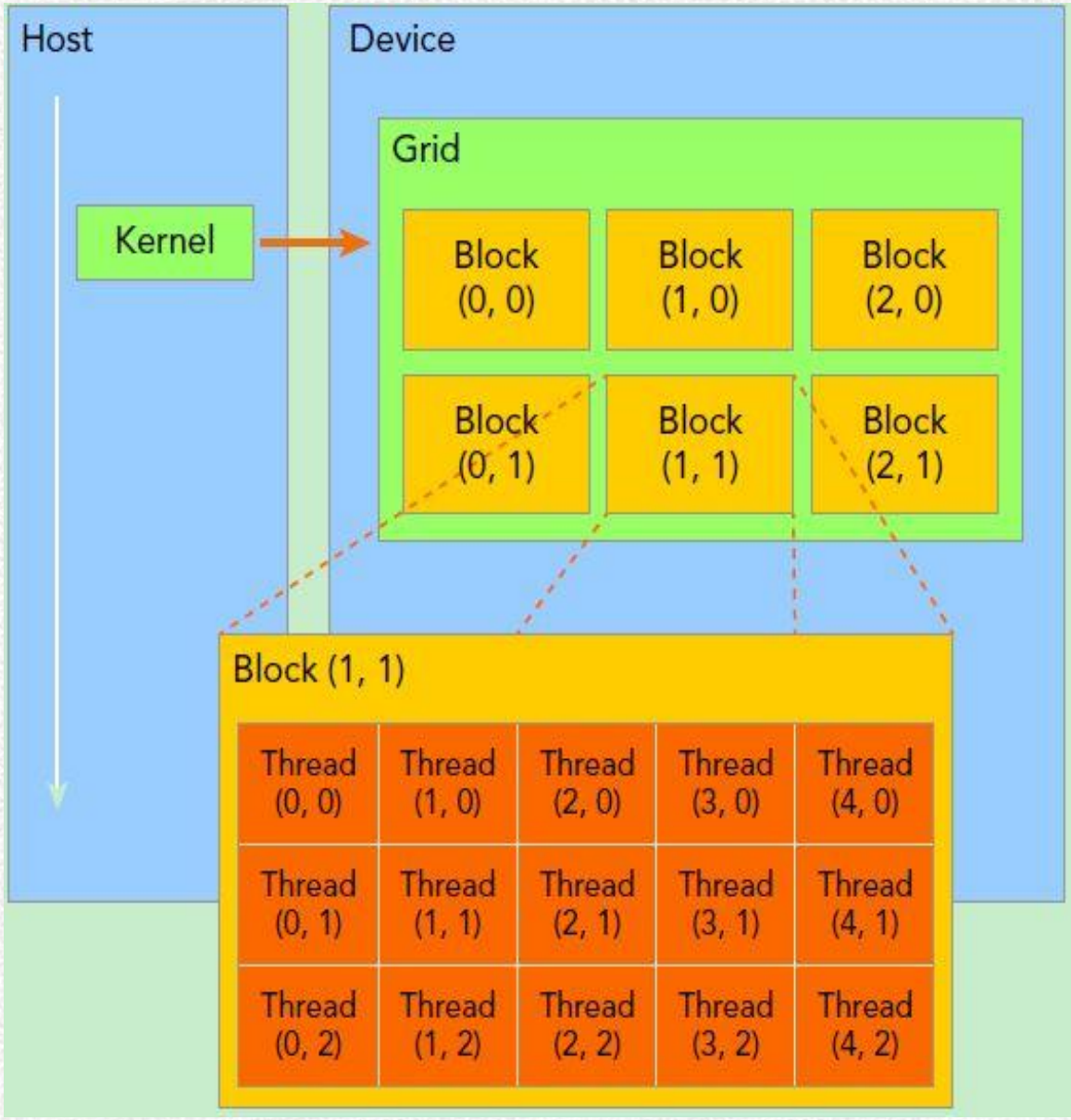
GPU Computing

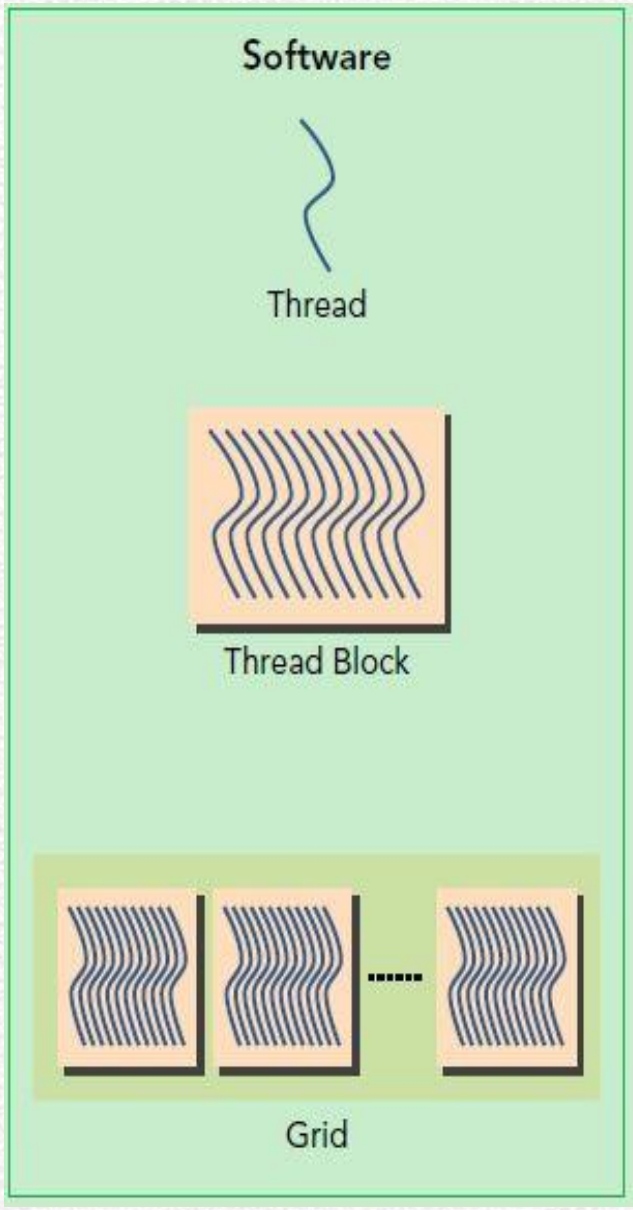


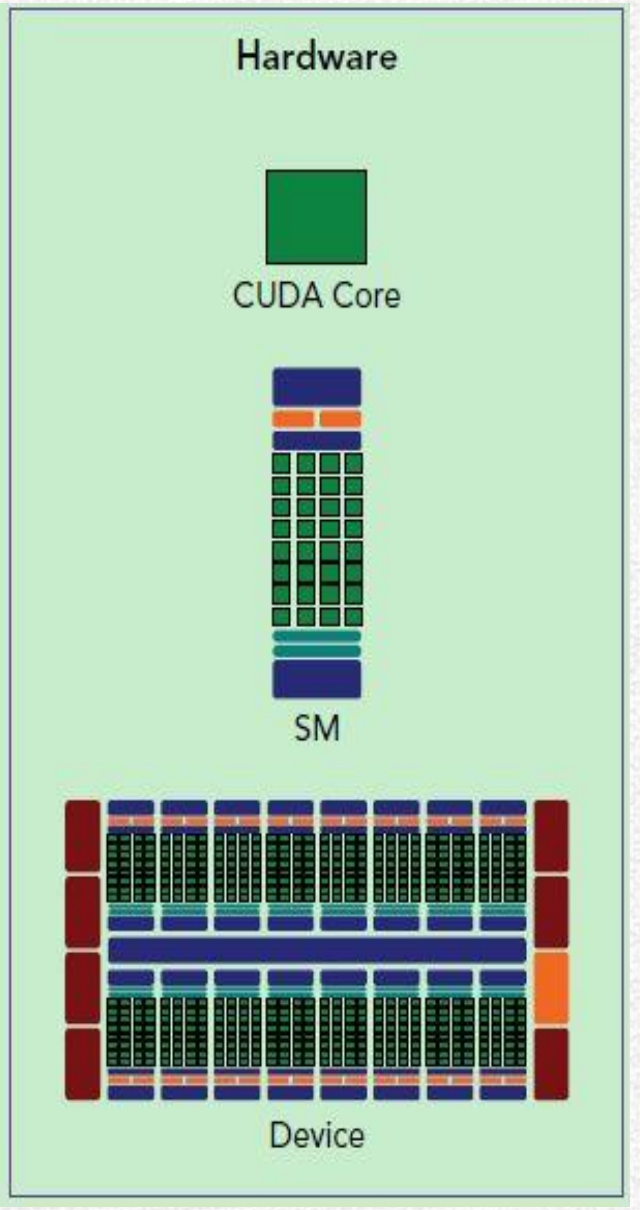
CUDA 程序执行与硬件映射

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

CUDA 程序架构以及硬件映射







GPU 流式多处理器

- kernel的线程组织层次，那么一个kernel实际上会启动很多线程，这些线程是逻辑上并行的，但是在物理层却并不一定。
- GPU硬件的一个核心组件是SM，英文名是 Streaming Multiprocessor (流式多处理器)。
- SM的核心组件包括CUDA核心，共享内存，寄存器等，SM可以并发地执行数百个线程，并发能力就取决于SM所拥有的资源数。
- 当一个kernel被执行时，它的gird中的线程块被分配到SM上，一个线程块只能在一个SM上被调度。
- SM一般可以调度多个线程块。那么有可能一个kernel的各个线程块被分配多个SM，所以grid只是逻辑层，而SM才是执行的物理层。

CUDA 内置变量

- 一个线程需要两个内置的坐标变量（blockIdx，threadIdx）来唯一标识，它们都是dim3类型变量，其中blockIdx指明线程所在grid中的位置，而threaIdx指明线程所在block中的位置:
- threadIdx包含三个值: threadIdx.x, threadIdx.y, threadIdx.z
- blockIdx同样包含三个值: blockIdx.x, blockIdx.y, blockIdx.z

- 逻辑顺序: X > Y > Z

dim3 grid(3, 2);
dim3 block(5, 3);

- 块: (0, 0), (1, 0), (2, 0), (0, 1), (1, 1), (2, 1)
- 线程: (0, 0), (1, 0), (2, 0), (3, 0), (4, 0), (0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (0, 2), (1, 2), (2, 2), (3, 2), (4, 2)

WARP 技术细节

- SM采用的是SIMT (Single-Instruction, Multiple-Thread，单指令多线程)架构，基本的执行单元是线程束 (warp)，线程束包含32个线程
- 线程同时执行相同的指令，但是每个线程都包含自己的指令地址计数器和寄存器状态，也有自己独立的执行路径。
- 程束中的线程同时从同一程序地址执行，但是可能具有不同的行为，比如遇到了分支结构，一些线程可能进入这个分支，但是另外一些有可能不执行，它们只能死等
- GPU规定线程束中所有线程在同一周期执行相同的指令，线程束分化会导致性能下降。
- 极小化命令的分化

性能优化

- 当线程块被划分到某个SM上时，它将进一步划分为多个线程束 (warp)，它才是SM的基本执行单元
- 因为资源限制，一个SM同时并发的线程束数是有限的。SM要为每个线程块分配共享内存，而也要为每个线程束中的线程分配独立的寄存器。所以SM的配置会影响其所支持的线程块和线程束并发数量。
- 网格和线程块只是逻辑划分，一个kernel的所有线程其实在物理层是不一定同时并发的。
- kernel的grid和block的配置不同，性能会出现差异，这点是要特别注意的。还有，由于SM的基本执行单元是包含32个线程的线程束，所以block大小一般要设置为32的倍数。

THANK YOU