CUDA執行模型

Streaming Multiprocessor

- Cores
- Shared memory/ L1 cache
- Register file
- LD/ST units
- Special function units
- Warp scheduler

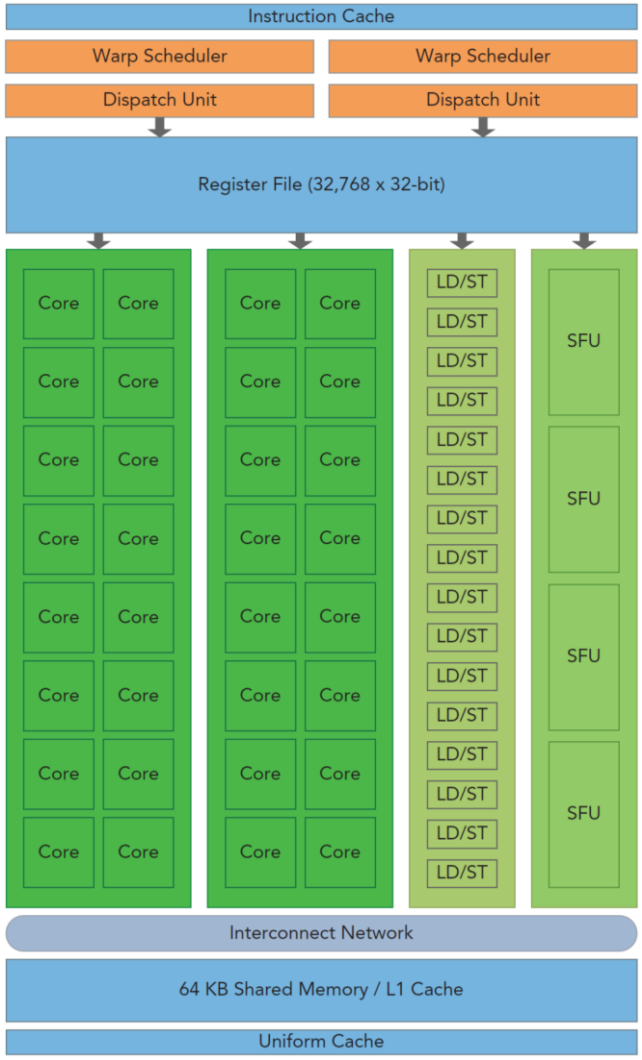

FIGURE 3-1

Streaming Multiprocessor

- Blocks分配给可用的SM
- Block内部的threads在对应SM上并发执行

SIMT

- Single Instruction Multiple Thread
- Warp: 一组32个thread
  - 在同一个block内
  - 每个thread执行相同指令
  - 一个warp内的thread可能有不同行为
    - 分支等待避免：分支粒度调整为32的整数倍
- 与SIMD的区别：每个thread有独立的
  - 指令地址计数器
  - 寄存器
  - 执行路径

片上资源分配

- Shared memory按block分配
- Registers按thread分配

- Active block: 已分配好计算资源的block
- Active warp: active block中的warp
  - Selected, eligible and stalled
  - 尽可能多的active warp来掩盖warp stall的延迟

确定grid和block大小的原则

- Block大小是warp大小(32)的整数倍
- 避免使用小的block(每个SM的并发warp数有限)
- 根据kernel的资源需求调整block大小
- Block数量远多于SM数量
- 通过实验确定最佳的大小参数

同步

- 系统级：host and device
  - cudaDeviceSynchornize()
- Block级：block内所有thread
  - __syncthreads()

并行优化

- 使用nvprof检测性能特征
- 避免分支等待
  - 重排数组下标，重排计算次序
- 循环展开
  - 更多并发操作
  - 避免reducing时warp间同步(也可用volatile变量)
  - 完全展开(循环次数已知)
  - Template函数(运行时确定，可移除部分if)

动态并行

- 在GPU上创建和同步新的kernel
- Parent grid和child grid共享global memory和constant memory，私有local memory和shared memory