NVIDIA

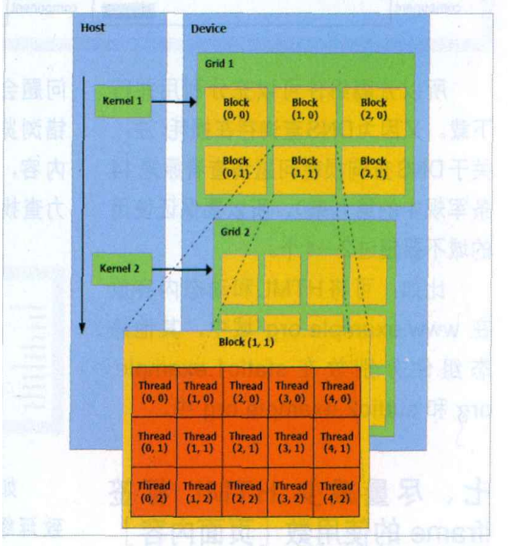
CUDA 编程模型

NVIDIA亚太区技术市场经理 邓培智

在CUDA的架构中，GPU可视为一个计算设备，是主机或者CPU的协处理器，用于处理高度并行的计算。GPU（或者称为“设备，device”）均具备自己的存储器（device memory，设备内存），可以并行地运行许多线程。在CUDA程序中，并行计算的部分可以被分离到一个被称为kernel（内核）的函数。在设备上许多线程执行同一个kernel。

CUDA的线程和CPU的线程有很大的不同。GPU线程是超级轻量的，创建线程的开销非常小，而且切换几乎是立即的。现在多核的CPU之需要有限的几个线程就可以使得CPU满负荷工作，但是GPU需要上千的线程才能够使之获得足够高的效率。

在设备中，执行内核的许多线程组织为线程块网格（grid）的方式。其组织如图所示：



线程快包含有一批线程SIMD（单指令多数据流）方式执行的线程，每个线程都有自己的ID，同时每个线程块也有自己的ID。在每个块内的线称可以相互协作：可以共享结果以节省计算；可以同步；线程块可以对共享内存访问，从而大大节省设备内存带宽的消耗。可以说线程协作是CUDA强大性能之一。不过，不同线程块内的线程不能进行协作。

将线程组织为线程块的方式会带来线程协作的降低，因为不同块之间的线程不能协作，但是好处是可以带来透明的硬件伸缩性。因为不同的设备有不同的并行能力。在并行度很高的设备上执行时很多块可以并行地执行；反之在较低并行度的设备上并行执行的块可以相对较少。

线程块和线程都有ID，其中线程ID可以是1D-3D的，3D的ID如Thread (x,y,z)；而线程块则可以是1D或者2D的。在图像处理等多维数据的场合，可以简化内存寻址。

在设备上执行的线程可以访问多种内存，包括设备上的DRAM和GPU芯片内置的寄存器和共享内存。下图是内存模型的示意图。线程、线程块和grid分别可以访问的内存如下：

线程可以读写寄存器

线程可以读写本地内存

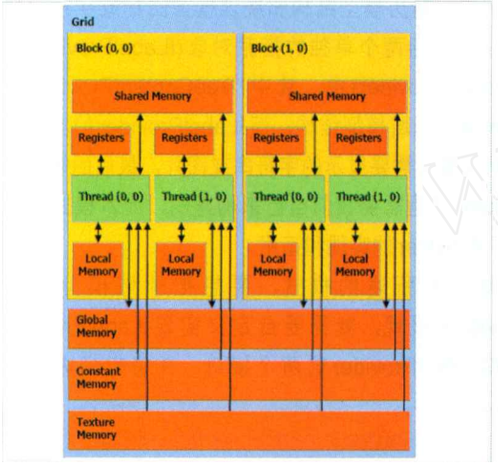
线程块可以读写共享内存

Grid可以读写全局内存

Grid对常量内存是只读的

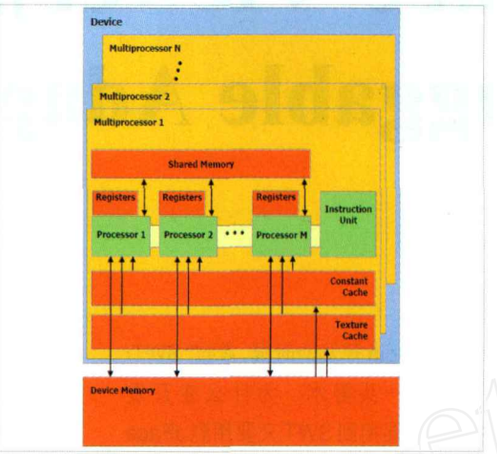
Grid对纹理内存是只读的

主机可以读写存储在设备DRAM中的全局内存、常量内存和纹理内存。



共享内存位于芯片内部，其速度等同于寄存器。其速度比没有高速缓存的全局内存快得多。将线程按照块的方式进行组织，在块内进行线程间的协作以及多次的计算时使用共享内存，可以大大提高计算和存储的效率，并且极大地减少所需设备DRAM的带宽。

在硬件层面上，设备或者说CUDA GPU包含有数量不一的Multiprocessor。每一个Multiprocessor里面有一组单指令多数据流架构的32位处理器。在每一个周期内，一个multiprocessor在被称为warp的一组线程上执行同样的指令，每一个warp包含的线程数量被称为warp size。程序员所面对的本地、全局、常量和纹理存储器实际在物理上都位于设备存储器（或者GPU DRAM）中，而在芯片内每个multiprocessor还包含有32位的寄存器（每个处理器均有一组）、一个芯片内的共享内存、一个只读的常量高速缓存以及一个只读的纹理高速缓存。常量高速缓存和纹理高速缓存可以加速常量存储器和纹理存储器的访问。CUDA GPU硬件结构的示意图如下：



在GPU程序执行的时候，每个grid的线程块被拆分到warp以内，每个线程块只能被一个multiprocessor执行，因此共享内存空间就固定在芯片内的共享内存内了。寄存器被分配给线程。如果一个kernel要求过多的寄存器则会出现启动失败。一个multiprocessor可以同时执行多个线程块，片内的存储资源则会被分配给这些并发的线程块中。如果线程块和线程减少对共享内存和寄存器的使用，就会使得并发的线程块数量增加，从而获得更高的并行度。因此在编程的时候需要注意节约这些资源的使用。

以上是CUDA软硬件模型的一些描述。如果我们对编程模型做个简洁的总结的话可以用下面这些公式：

设备（device）= GPU = 一组multiprocessor

Multiprocessor = 一组处理器以及共享内存

Kernel = GPU程序

Grid = 执行kernel的线程块的阵列

线程块 = 执行一个kernel并能够通过共享内存进行通信的一组SIMD线程

NVIDIA目前推出了一系列的支持CUDA的GPU，这些GPU具备不同的并行计算能力，每种GPU都有不同数量的multiprocessor。这些GPU的架构是非常类似的，但是每种GPU的multiprocessor具备的内部资源以及wrap size等参数可能会有区别。详细的情况可以进入http://www.nvidia.com/object/cuda_home.html去查阅。更多有关CUDA编程的信息请参阅CUDA编程指南。