GPU 学习笔记

cudaDeviceProp 结构中设备的属性：**Nvidia CUDA Programming Guide**

编译器	函数 kernel 被交给编译设备的编译器 main 函数被交给主机的编译器	
设备内存的分配与释放 设备内存指针 cudaMalloc() cudaFree()	设备上执行任何操作都需要分配内存 可以将cudaMalloc()分配的指针传递给在设备上执行的函数。 可以在设备代码中使用cudaMalloc()分配的指针进行内存读/写操作。 可以将cudaMalloc()分配的指针传递给在主机上执行的函数。 不能在主机代码中使用cudaMalloc()分配的指针进行内存读/写操作。	
	1.	
设备内存的访问	2. 在设备代码中使用设备指针 3. Host 代码中调用 cudaMemcpy()	

设备上能够执行的函数	编写：__global__修饰符 -> 被称为核函数
	调用：特殊的尖括号语法<<<>>>
	`dim3 blocks(DIM/16,DIM/16);`
	`dim3 threads(16,16);`
	`kernel<<< blocks,threads>>>(`	

	是每个 block 都有一块共享内存吗？	

Device Memory

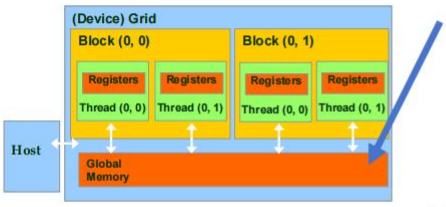
just set and use

API 级别的粒度 有句话说这是软件级别的设置，Q：那不同的设置有什么区别？速度上

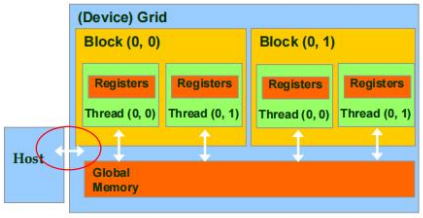
		内置变量	最大值
thread	threadIdx	threadDim	512
block	blockIdx	blockDim	65535
grid	gridIdx		

修饰符

__global__
__device__



- cudaMalloc()
  - Allocates an object in the device global memory
  - Two parameters
    - Address of a pointer to the allocated object
    - Size of allocated object in terms of bytes
- cudaFree()
  - Frees object from device global memory
  - One parameter
    - Pointer to freed object



- cudaMemcpy()
  - memory data transfer
  - Requires four parameters
    - Pointer to destination
    - Pointer to source
    - Number of bytes copied
    - Type/Direction of transfer
  - Transfer to device is synchronous with respect to the host

This is an equivalent way to express the ceiling function.

```
dim3 DimGrid((n-1)/256 + 1, 1, 1);
```

卷积