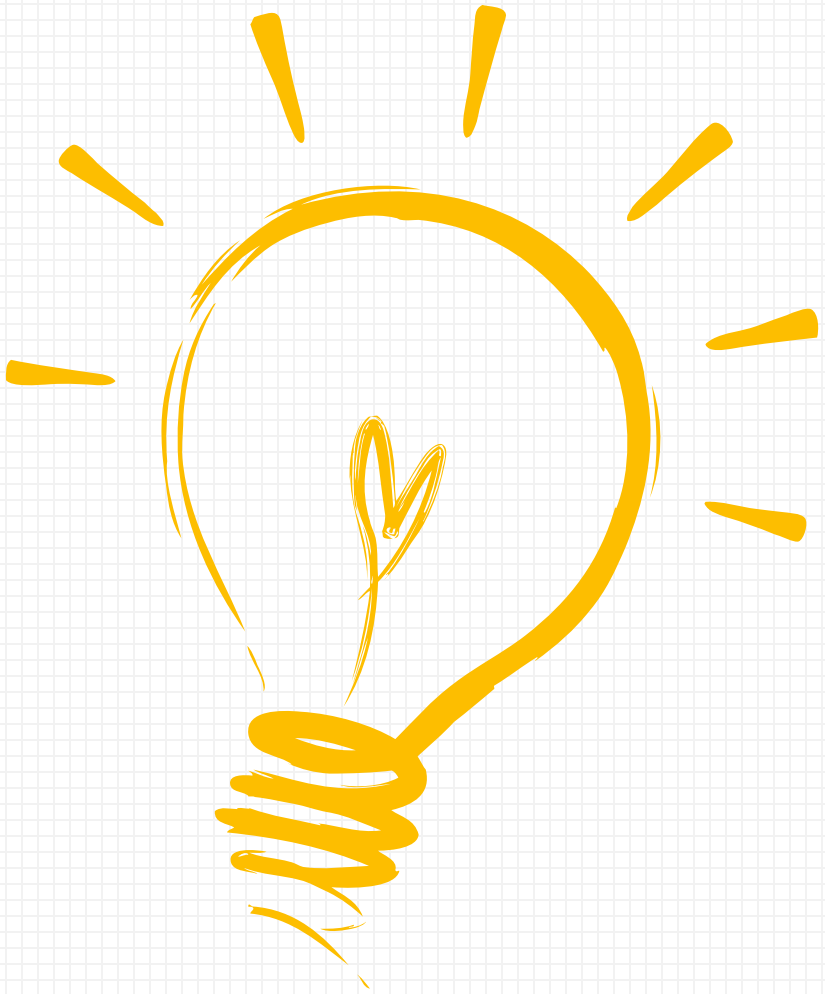
线程全局索引计算方式

CUDA并行编程系列课程

主讲：权双

CONTENTS



一维网格 一维线程块

★ 定义grid和block尺寸:

dim3 grid_size(4);

dim3 block_size(8);

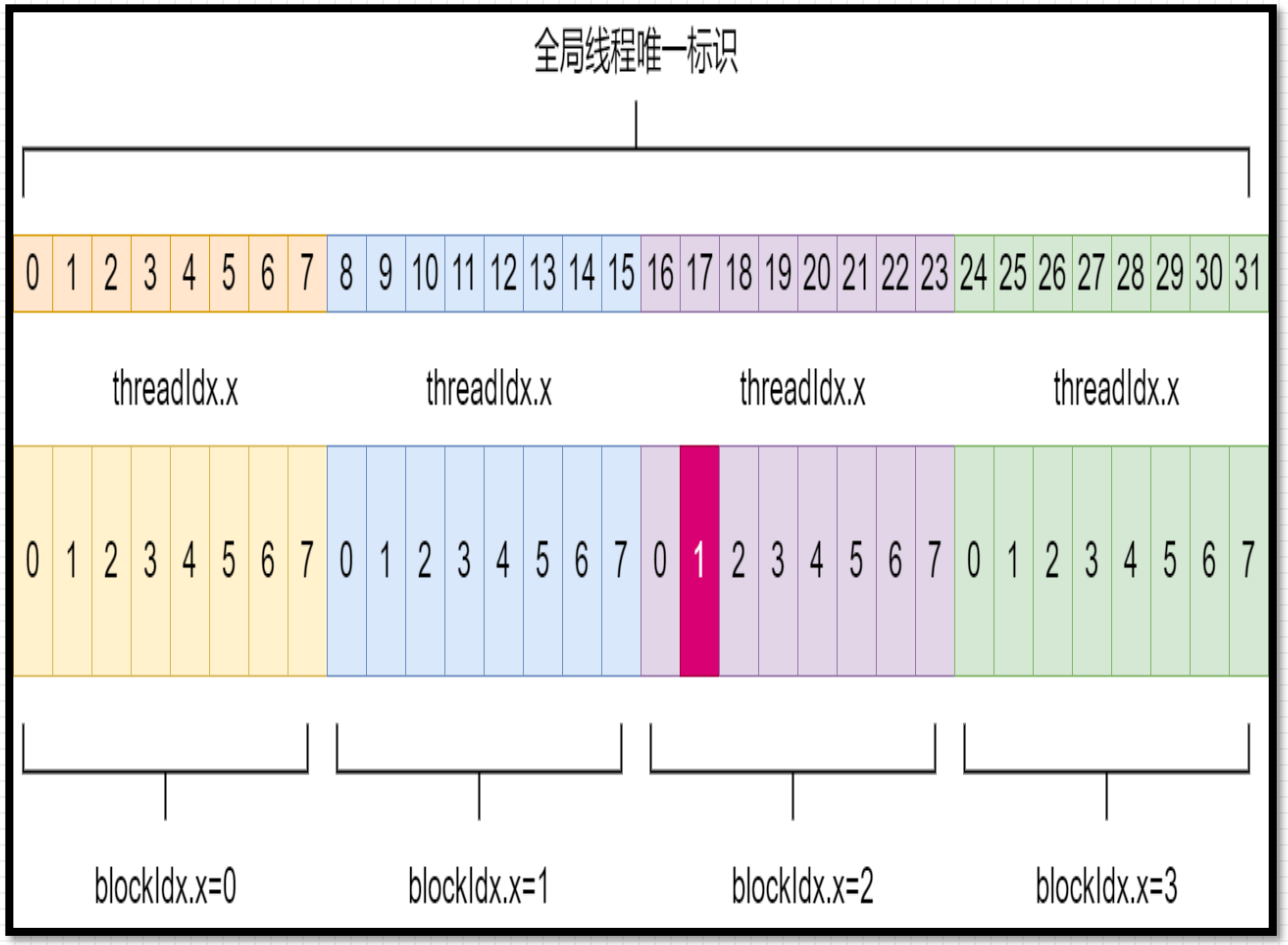
★ 调用核函数:

kernel_fun<<< grid_size, block_size >>>(…) ;

★ 具体的线程索引方式如图所示，blockIdx.x从0到3，threadIdx.x从0到7。

★ 计算方式:

int id = blockIdx.x * blockDim.x + threadIdx.x;



二维网格 二维线程块

★ 定义grid和block尺寸:

dim3 grid_size(2, 2);

dim3 block_size(4, 4);

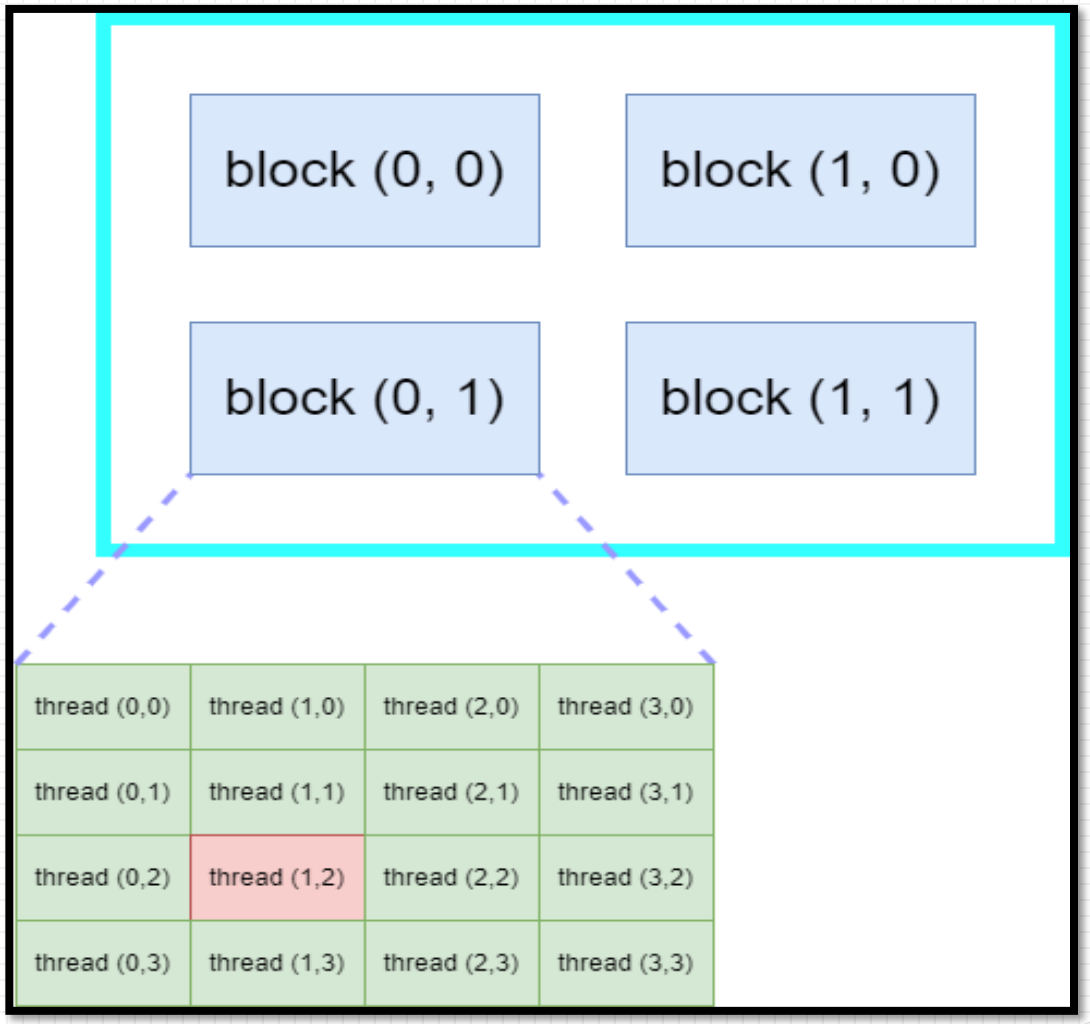
★ 调用核函数:

kernel_fun<<< grid_size, block_size >>>(…) ;

★ 具体的线程索引方式如图所示， blockIdx.x和blockIdx.y从0到1， threadIdx.x和threadIdx.y从0到3。

★ 计算方式:

int blockId = blockIdx.x + blockId.y * gridDim.x;
int threadId = threadIdx.y *blockDim.x + threadIdx.x;
int id = blockId * (blockDim.x * blockDim.y) + threadId;



三维网格 三维线程块

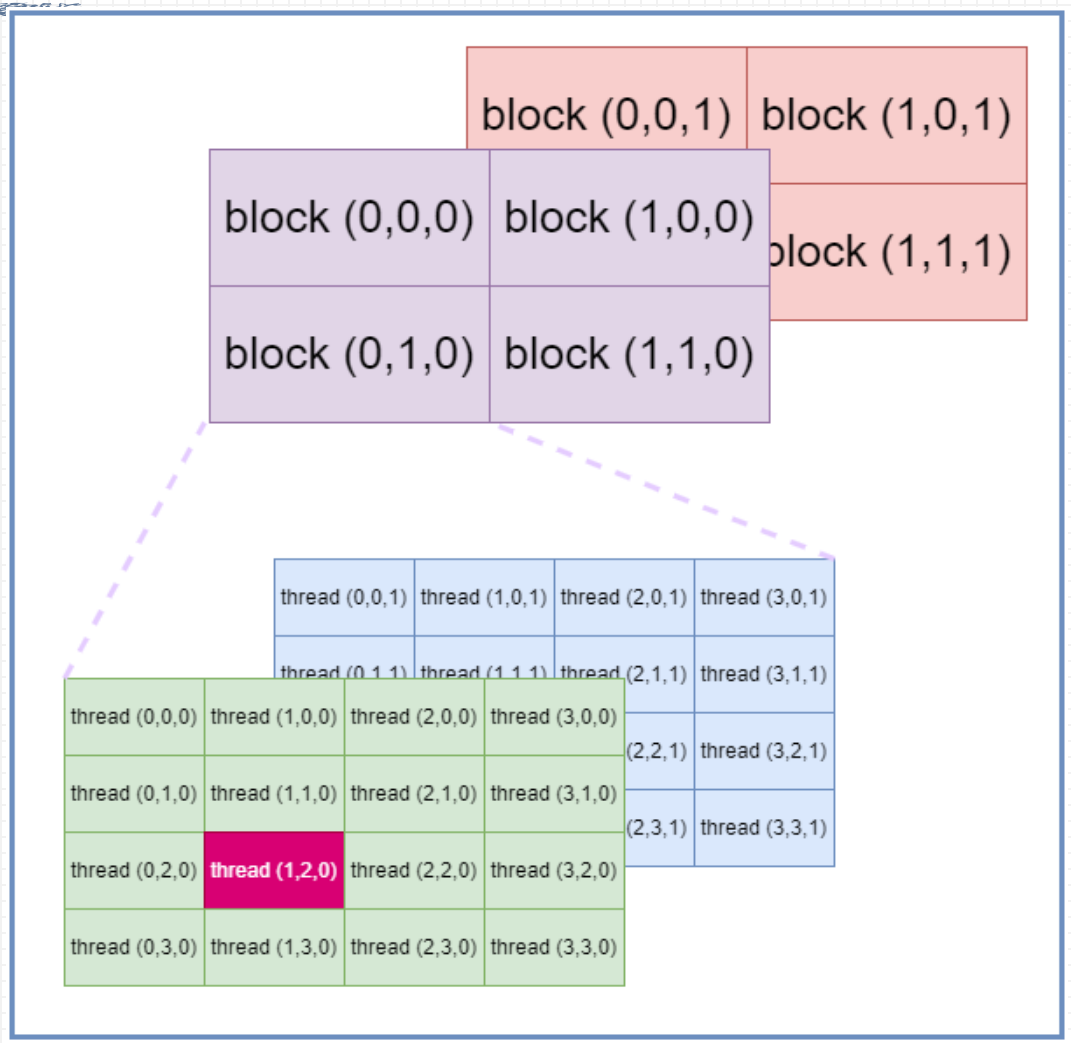
★定义grid和block尺寸:

dim3 grid_size(2, 2, 2);　　dim3 block_size(4, 4, 2);

★调用核函数:

kernel_fun<<< grid_size, block_size >>>(…) ;

★具体的线程索引方式如图所示，blockIdx.x、blockIdx.y和blockIdx.z从0到1， threadIdx.x、 threadIdx.y从0到3，threadIdx.z从0到1。



★计算方式:

```
int blockId = blockIdx.x + blockIdx.y * gridDim.x
                + gridDim.x * gridDim.y * blockIdx.z;
int threadId = (threadIdx.z * (blockDim.x * blockDim.y))
                + (threadIdx.y * blockDim.x) + threadIdx.x;
int id = blockId * (blockDim.x * blockDim.y * blockDim.z) + threadId
```

不同组合形式

★ 一维Grid 一维Block：

```
int blockId = blockIdx.x;
int id = blockIdx.x *blockDim.x + threadIdx.x;
```

★ 一维Grid 二维Block:

```
int blockId = blockIdx.x;
int id = blockIdx.x * blockDim.x * blockDim.y + threadIdx.y * blockDim.x + threadIdx.x;
```

★ 一维Grid 三维Block

```
int blockId = blockIdx.x;
int id = blockIdx.x * blockDim.x * blockDim.y * blockDim.z
                + threadIdx.z * blockDim.y * blockDim.x
                + threadIdx.y * blockDim.x + threadIdx.x;
```

不同组合形式

★ 二维Grid 一维Block：

```
int blockId = blockIdx.y * gridDim.x + blockIdx.x;
int id = blockId * blockDim.x + threadIdx.x;
```

★ 二维Grid 二维Block:

```
int blockId = blockIdx.x + blockIdx.y * gridDim.x;
int id = blockId * (blockDim.x * blockDim.y) + (threadIdx.y * blockDim.x) + threadIdx.x;
```

★ 二维Grid 三维Block

```
int blockId = blockIdx.x + blockIdx.y * gridDim.x;
int id = blockId * (blockDim.x * blockDim.y * blockDim.z)
                 + (threadIdx.z * (blockDim.x * blockDim.y))
                 + (threadIdx.y * blockDim.x) + threadIdx.x;
```

不同组合形式

★ 三维Grid 一维Block：

```
int blockId = blockIdx.x + blockIdx.y * gridDim.x + gridDim.x * gridDim.y * blockIdx.z;
int id = blockId * blockDim.x + threadIdx.x;
```

★ 三维Grid 二维Block:

```
int blockId = blockIdx.x + blockIdx.y * gridDim.x + gridDim.x * gridDim.y * blockIdx.z;
int id = blockId * (blockDim.x * blockDim.y) + (threadIdx.y * blockDim.x) + threadIdx.x;
```

★ 三维Grid 三维Block

```
int blockId = blockIdx.x + blockIdx.y * gridDim.x + gridDim.x * gridDim.y * blockIdx.z;
int id = blockId * (blockDim.x * blockDim.y * blockDim.z)
         + (threadIdx.z * (blockDim.x * blockDim.y))
         + (threadIdx.y * blockDim.x) + threadIdx.x;
```

THANKS

谢谢聆听