Apuntes programación CUDA

Mauricio Vanzulli

April 27, 2021

1 Clase 1 CUDA

Características del device:

- Es un coprocesador de la CPU
- Posee una memoria DRAM
- Ejecuta muchos threads en paralelo.
- Cada mutli-procesador procesa un bloque con un programa único (kernel) en muchos hilos. Cada CUDA-core procesa muchos hilos, uno a la vez.
- Este paradigma de programación recibe la sigla SPMT (single program multiple threads).
- Cada kernel ejecuta un array de hilos, es importante el identificador de hilo respecto al dato al que se le quiere ejecutar el kernel.

Algoritmo básico de programación

1. Instrucciones en el host.
2. Enviar los datos al device.
3. Procesa en GPU
4. Recuperar los datos de la GPU.
5. Continuar procesamiento en el host.

Existe determinada jerarquía de threads. Una grilla en 3D agrupa un conjunto de bloques y dentro de cada bloque se tienen múltiples hilos también en 3D. No se puede asumir a priori que el bloque 1 se ejecute antes del 2.

Las funciones en CUDA

Dominio de funciones	*Ejecuta en:*	*Se invoca desde:*
__device__	device	device
__global__	device	host
__host__	host	hostt

Función	*Variables de entrada:*	*Proposito:*
dim3 DimGrid	(Cant_Bloq_x,Cant_Bloq_y,Cant_Bloq_z)	Crea una grilla de esas dimensio de bloques
dim3 DimBlock	(Cant_Hilos_x, Cant_Hilos_y, Cant_Hilos_z)	Crea las dimensiones de los threads en cada bloque
KernelFunc <<<DimGrid,DimBlock,SharedMemBytes>>>(...)	Input Kernel, grilla, y memoria compartida	Ejecuta el kernel en device

Se disponen de distintos espacios de memoria:

- __global__ : memoria global en el host
- __device__ : memoria global en device
- __shared__ : reside en la memoria compartida host-device (se usa para allocamiento dinámico)
- __constant__ reside en momoria cosntante del device (allocamiento estático)

Localizadores de hilos y bloques:

- threadIdx :Ubicación de ese thread y se accede con .x .y .z
- blockIdx :Ubicación del bloque en el que me encuentro y se acccede con .x .y .z
- blockDim : Tamaño del bloque (en cantidad de hilos) y se accede con .x .y .z
- gridDim :Tamaño de la grilla (en cantidad de blques)y se accede con .x .y .z

Funciones intrínsecas al GPU

- cudaDeviceSynchronize :Sincroniza todos hilos en el device y se ejecuta desde el host.
- __syncthreads :Permite sincronizar los threads de un mismo bloque.

Reservar memoria en la tarjeta y transferir datos es fundamental para comunicar las datos y sus procesamientos, esto es bastante costoso en términos computacionales. Me interesa comunicar host-cpu (se ejecutan desde el CPU):

- cudaMalloc(Puntero,Tamaño de memoria) Reserva memoria en la global memory de la GPU
- cudaFree (Puntero) libera el espacio reservado

- cudaMemcpy (Puntero de destino, puntero origen,numero de bytes a copiar, Tipo de transferencia) . Los diferentes tipos de transferencia son: Host 2 Device (cudaMemcpyHostToDevice), Device 2 Host(cudaMemcpyDeviceToHost). Además están (Host 2 Host) y Device 2 Device.
- cudaMemSet (Variable a iniciar,0 o 1, sizevector): Setea el inicio del vector en el device.