Parallel Programming Test 2

Name: 刘康来   Student ID: 2019011777

May 2021

(*Please hand in a PDF version.*)

1 What are the differences between CPU, GPU and GPGPU? Why does GPGPU sometimes have better performance than CPU? Also, how to write a good GPU program with CUDA language? (no more than 150 words)

Ans:

The CPU is composed of just a few cores with lots of cache memory that can handle a few software threads at a time.
The GPU is composed of hundreds of cores that can handle thousands of threads simultaneously.
The GPGPU is used for specialized programming and equipment designs with massive parallel processing of non-specialized calculations.
Because GPGPU have specialized designs for specialized work.
To write a good CUDA code, we should know the hardware well. To use the shared memory properly.

2 Please describe the steps (e.g. cudaMalloc, cudaMemcpy, kernel launch) required to write a CUDA program. (no more than 100 words)

Ans:

1. cudoMalloc first, than copy the data from CPU to GPU use cudaMemcpy(Md, M, size, cudaMemcpyHostToDevide).
2.run the kernel on device from host using MatMulKermnel«<dimGrid, dimBlock»>().
3. Copy the data from CPU to CPU use cudaMemcpy(M, Md, size, cudaMemcpyDeviceToHost).
4. Free the device memory using cudaFree and free the host memory using free.

3 How many kinds of memory can be used in CUDA language to compute on GPU? What are their names and functions? (no more than 100 words)

Ans:

Global, local, texture, constant, shared and register memory.
Register memory is visible only to the thread that wrote it and lasts only for the lifetime of that thread.
Local memory is same as register memory, but performs slower.
Shared memory is visible to all threads within that block and lasts with the block.
Global memory is visible to all threads within the application (including the host), and lasts for the duration of the host allocation.
Not about:
(Constant and texture memory used for very specific types of applications. Constant memory is used for data that will not change over the course of a kernel execution and is read only. Using constant rather than global memory can reduce the required memory bandwidth, however, this performance gain can only be realized when a warp of threads read the same location.Similar to constant memory, texture memory is another variety of read-only memory on the device.

When all reads in a warp are physically adjacent, using texture memory can reduce memory traffic and increase performance compared to global memory.)

4 Why we need shared memory? Can all threads read and write to the same partition of shared memory? Which threads can access the same partition of shared memory, and why? (no more than 100 words)

Ans:
Shared memory is faster than Global memory, and can be got by more threads.
No. Shared memory is divided into logical banks. Successive sections of memory are assigned to successive banks. Each bank services only one thread request at a time, multiple simultaneous accesses from different threads to the same bank result in a bank conflict (the accesses are serialized).
Threads can access data in shared memory loaded from global memory by other threads within the same thread block.

5 What should be done if the matrix size is not exactly divisible into a thread block dimension, when we write a CUDA GEMM code? For example, the size of matrix A is 300*200, the size of matrix B is 200*400, and the thread block dimension is 16*16. You are encouraged to draw a figure to describe your method. (no more than 100 words)

Ans:
dim3 dimGrid((n / dimBlock.x) + (n % dimBlock.x != 0), (m / dimBlock.y) + (m % dim-Block.y != 0));

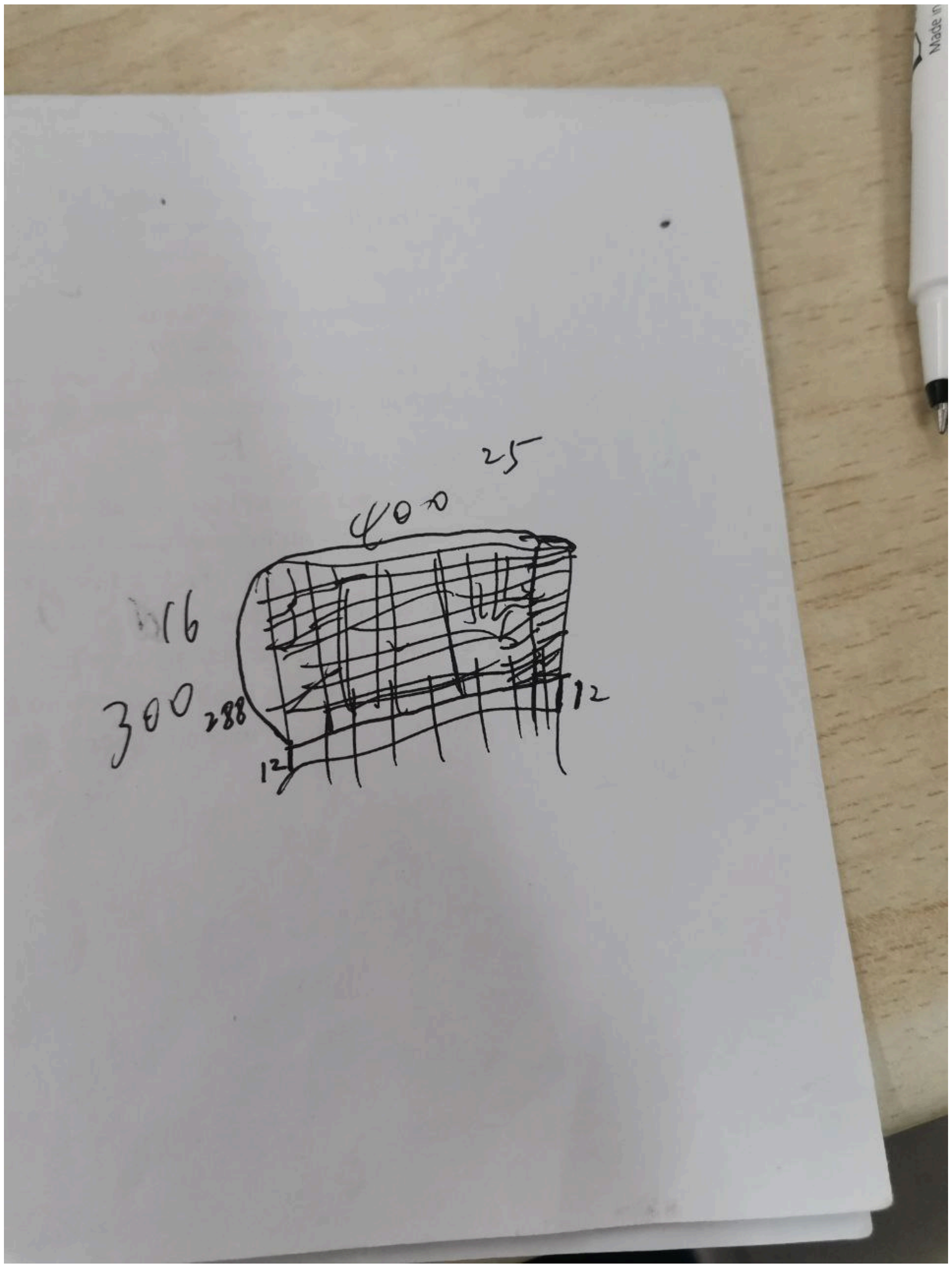

图 1: example