CSCI-GA.3033-004

Graphics Processing Units (GPUs): Architecture and Programming

Fall 2015 – (90 minutes)

NAME: **ID:**

- The exam is open book/notes.
- If you have to make assumptions to continue solving a problem, state your assumptions clearly.
- You answer on the question sheet. You can use extra white papers if you want.

1. [1 point] We know in CUDA that commands in a stream (e.g. kernel launch, data movement between host and device, etc) are executed in order. Why is this restriction, given that it may lead to some performance loss?

Because commands in a stream usually depend on each other (moving data from device to host need to wait till the kernel is done, etc …) so they need to be in-order to ensure correctness. If they are independent, then they can be issues in different streams for parallelism.

2. [2 points] We have seen that if-else may lead to branch divergence in a warp due to lockstep execution of instructions. Now, suppose there is a thread that has and *if* without else. Can this also lead to performance loss in some cases? If yes, explain a scenario where there is performance loss. If no, explain why not. No need to write full code, just explain.

Yes, it can lead to some performance loss if some threads in the warp have a true condition and others do not. In that case, those who have false, must wait till the others finish the if-part, leading to performance loss. However, it is not as severe as the if-else.

3. In OpenCL there is only one queue between the host and each device. So we cannot have several queues between the host and device like streams in CUDA.

a. [2 points] Does this restrict the performance of OpenCL? Justify.

No, because the task queue can be configured to be out-of-order, hence ensuring some parallelism.

b. [2 points] Will we gain any performance in OpenCL if we allow multiple queues between the host and the device? If yes, give a scenario where multiple queues give better performance. If no, explain why not.

No, we will not. Actually we can lose some performance due to the extra work in managing several task queues instead of one.

4. [2 points] Beside overlapping data-transfer and computation, state two other scenarios where streams are useful.

- Executing two different kernels in parallel

- Streams ensure correctness when commands of a single stream are executed in order.

5. [2 points] State two characteristics of a problem that makes it a good candidate for GPU instead of CPU.

- Massive data parallelism

- Compute bound

6. [3 points] State three reasons you may want to have several kernels instead of one big kernel.

- If you need to synchronize among threads of different blocks

- If the different kernels require different geometry

- If you want smaller threads with smaller resources to ensure parallelism among blocks in the same SM.

7. [4 points] Suppose NVIDIA decides to have larger warps in their future GPUs. Give two advantages of doing so, and two disadvantages.

Advantages:

- More opportunities for memory coalescing

- Potentially more parallelism

Disadvantages

- Higher probability of thread divergence

- Putting more restrictions on the programmer in choosing block size to be a multiple of bigger number

8. We have discussed a lot the importance of memory coalescing. Also we said that having an L2 cache (servicing all the SMs) helps in global memory coalescing.

a. [1 point] How does L2 helps in memory coalescing?

In an L2 miss, a cache block is fetched from the global memory. A cache block is coalesced as it consists of continuous bytes from memory.

b. [1 point] Does the existence of L2 mean that the programmer does not need to pay attention to global memory access to be coalesced? Explain.

No, still the program needs to be aware of memory access pattern because accessing separated memory locations means the L2 cache may need to fetch several blocks to satisfy the requirements, which is not coalesced.