Graphics Processing Units (GPUs): Architecture and Programming
Homework 1 Solutions

(total: 30 points)

1. [8 points: 1 for each Y/N and 1 for each justification]

Part	Resource (Y/N)	Reason
L1 Cache	N	- We cannot determine the amount of cache needed beforehand.
SP	N	- Can be assigned and re-assigned to different warps
Global Memory	N	- Not in SM
L2 cache	N	- We cannot determine the amount of cache needed beforehand. - Not in SM

2. [2 points] When the block is assigned to an SM, it is divided into a set of warps of 32 threads each. If the number of threads in the block is not divisible by 32, the last warp will have fewer than 32 threads. Then one or more warps, depending on the number of SPs, are assigned for execution. Then the following warps will execute once the once before are done or are waiting for a long latency operation.

3. [8 points]
   a) Yes, each core can be assigned a subset of the numbers and check the existence of the number in them. This is done in parallel.
   b) Cannot be done on GPU because the operations are dependent on each other.
   c) The problem size is not big enough to ensure enough parallelism to overcome communication overhead.
   d) Yes, it is a highly parallel operation and we have enough independent computations. Moreover, the problem size is big enough to ensure enough parallelism.

4. [2 points]
   - This will put a lot of pressure on that memory as it cannot serve all the cores in parallel. This memory will be a bottleneck that may affect scalability if we want to add more CPUs and/or GPUs.
   - GPU needs a memory optimized for bandwidth while CPU needs memory optimized for latency.

5. [2 points] Because in order to get good performance from SIMD we need to have large number of processing units. MIMD uses full-fledged cores as processing units. Those full-fledged cores are big in size. Hence, we cannot put a large number of them on-chip. However, in SIMD chips (e.g. GPUs), the processing units are just execution units not full-fledged cores. So, we can pack hundreds of them in the chip.

6. [1 points] Even though GPUs may consume/dissipate more power than traditional multicore, but, for GPU friendly application, they achieve much higher performance. Hence, GPUs have

better performance per Watt (i.e. better performance for the same amount of power) than multicore.

7. [1 point] For efficient programming, threads executing on SPs must be able to communicate and synchronize. However, with that large number of SPs in current GPUs, it will be very expensive to interconnect all of them together. Therefore, a solution is to group every few SPs together.

8. [6 points] For each one of the following statements, answer with Yes if it is correct or No if it is wrong.

a) **N** Two threads from two different blocks assigned to the same SM can see and access each other's shared memory.
b) **Y** We can have two blocks from two different kernels in the same SM at the same time.
c) **N** If we have 8 SMs in a GPU, then at most 8 blocks can be executing in parallel. (If an SM has enough resources, it can execute more than one block at a time.)
d) **N** If we have 64 SPs in SM then a block can have at most 64 threads.
e) **N** If a piece of code is computation intensive, with similar computations, and independent computations then we are sure to get high performance from using GPUs. (The problem size needs to be big enough to justify the overhead of sending the data to the device.).
f) **N** Two blocks assigned to the same SM must be of the same size (in terms of number of threads). (The two blocks may be coming from two different kernels.).