FAST NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

KARACHI CAMPUS

Quiz 2 (Section A) Fall 2021 (December 16, 2021)

Parallel and Distributed Computing (CS 3006)

Max. Points: 70 points.

Student Roll No: _____

Note: Illustrate means drawing a diagram and labelling it in detail only. Textual details are not required.

Q1

Why we need GPU as a compute accelerator? Don't you think OpenMP can address multi-threaded parallelism efficiently? Explain.

- GPU is a seperated device in addition to CPU
- Manage Threads in ~~hardware~~ hardware.
- 1000 of cores to increase the Throughput of data intensive computations

↳ - Runs on CPUs which have limited cores.
- OpenMP threads managed by O/S running on CPU along with other application. Context Switch is costly.
- CPU cores are general purpose, doesn't have specialized hardware e.g Integer, Floating point units.

Assume a grid of threads in execution on a NVIDIA GPU. Explain the GPU memory model offered and how it can be used by the programmer to write code which optimized execution time on GPU.

CUDA Threads —maps on→ Streaming Processor —uses→ Registers + local Memory
Cuda Block → Streaming Multi-proces → Shared Memory available to all Threads of Block.
Cuda Grid → Global Memory + Constant (Read only). Memory available to all blocks & Threads.

Section 5.2.
Figure 5.2.

Define a computation problem where utilizing GPU computation will significantly slows down execution time.

*. Matrix multiplication of dimension 4×4, where each element in an integer.

A GPU execution of this will require data movement time. which will increase overall time of completion of program.

Explain execution of thread on a streaming multiprocessor (SM). How SM could hide the latency of global memory access?

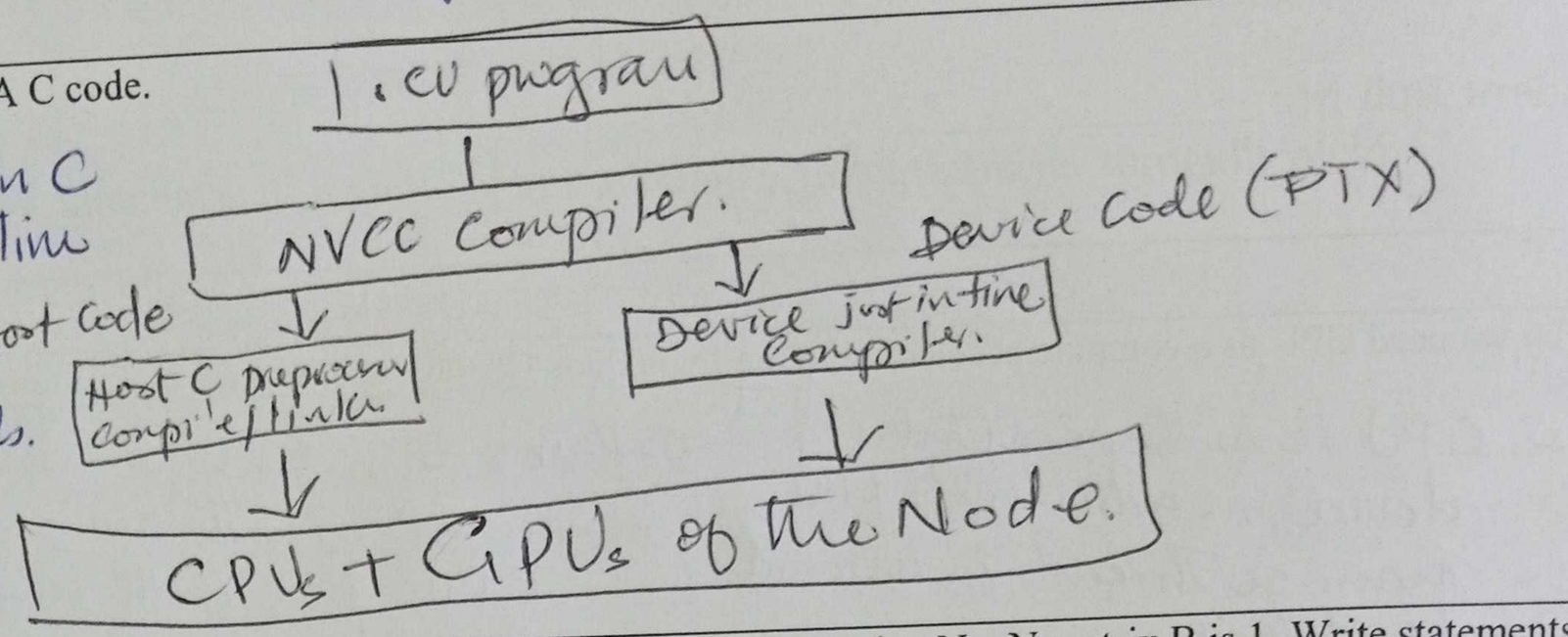
CUDA Blocks maps to a SM. Threads are divided into Warps of 32 threads. Warp is a unit of Scheduling. Warp are executed based on SIMD model and single instruction is fetched and executed for all threads in the warp to process different portions of data. Section 4.7 Figure 4.14

Many threads blocks are mapped to each SM. and many warps wait for execution. Ready warps are scheduled for execution. This mechanism to fill ~~wait~~ the latency time of operation ~~with~~ ~~wait~~ by scheduling other warps hide the latency of delay experience by accessing global memory.

Q2.

Illustrate compilation and execution of a CUDA C code.

CUDA extention embedded in C code will be called at runtime to move data to-from GPU and schedule kernal code calls afterwards.



Section 3.2
Figure 3.2

Write GPU only code to set elements of an N x N matrix A to 0, if the corresponding element in another N x N matrix B is 1. Write statements to run your GPU code?

Assume execution is performed using a grid size of 1 and block size of N x N.

```
__global__ void MatProcess(double A[N][N], double B[N][N]) {
    int tx = threadIdx.x; ty = threadIdx.y;
    if (B[tx][tx] == 1) A[tx][ty] = 0;
}
```

```
int numBlocks = 1;
dim3 threads_per_block(N, N);
MatProcess<<<numBlocks, threads_per_block>>>(A, B);
```
→ Inserted in main();

Explain the execution of the above code. Assume the value of N is 1000.

One thread block with (1000, 1000, 1) threads. (x, y, z)

Based on their thread index, threads are grouped into warps of 32 and executed on every available SM.