

Unit 10 - Week 8

Course outline

How does an NPTEL online course work?

Week 0 Assignment 0

Week 1

Week 2

Week 3

Week 4

Week 5

Week 6

Week 7

Week 8

● Lecture 36 : Thread execution in CUDA program - scheduling and memory access

● Lecture 37 : Thread execution in CUDA program (continued)

● Lecture 38 : Matrix multiplications in CUDA

○ Lecture 39 : OpenACC programming for GPU-s

● Lecture 40 : Hybrid parallelization and exascale computing

● Lecture material of Week 8

○ Quiz : Week 8 Assignment 8

● Week 8 Feedback Form

Download Videos

Detail Solution

Live Interactive Session

Text Transcripts

Week 8 Assignment 8

The due date for submitting this assignment has passed.
As per our records you have not submitted this assignment.

Due on 2020-11-11, 23:59 IST.

1) Which of the following is the scheduling unit of threads in a GPU streaming multiprocessor? 1 point

- a. Blocks
- b. Grids
- c. Maximum number of threads in a block in x direction
- d. Warp

- a.
 b.
 c.
 d.

No, the answer is incorrect.

Score: 0

Accepted Answers:

d.

2) The total number of threads active in a GPU at any instant is fixed by its hardware and known as warp size. 1 point

- a. True
- b. False

- a.
 b.

No, the answer is incorrect.

Score: 0

Accepted Answers:

b.

3) Why will one get typically small speed-up in a GPU code if all the threads are launched in a single block 1 point

- a. Most of the SM-s will be inactive
- b. All threads will be executed serially
- c. There will be much contention in accessing global memory
- d. Cache coherence issues

- a.
 b.
 c.
 d.

No, the answer is incorrect.

Score: 0

Accepted Answers:

a.

4) How does a GPU perform context switching to get better parallelism? 1 point

- a. Using better control over the hardware
- b. Through CPU kernel calls using streams
- c. By using large number of registers
- d. None of the above

- a.
 b.
 c.
 d.

No, the answer is incorrect.

Score: 0

Accepted Answers:

c.

5) Consider the following code snippet. Why will this kernel give a low speed-up? 1 point

```
__global__ void branchTest_kernel( float* a){
    int tx = threadIdx.x;

    if(tx==0){
        a[1] = a[0] + 1; (a) // or tx==1
    }else if(tx==1){
        a[0] = a[1] + 1;; (b) // or tx==0
    }
}
```

- a. Smaller number of threads
- b. High compute to global memory access
- c. Use of a smaller number of registers
- d. Thread divergence

- a.
 b.
 c.
 d.

No, the answer is incorrect.

Score: 0

Accepted Answers:

d.

6) Shared memory is small on-chip memory shared by all threads in a 1 point

- a. Warp
- b. Block
- c. Grid
- d. Stream

- a.
 b.
 c.
 d.

No, the answer is incorrect.

Score: 0

Accepted Answers:

b.

7) In a matrix-vector product using CUDA kernel, how a thread can be assigned to work on a particular row of the matrix 1 point

- a. Programmer specifies a separate pointer for the thread and map it with the matrix
- b. Threadidx function can find local thread id and map it to the matrix row
- c. Global thread id using threadidx and blockidx and blockdim is obtained and this id points to the matrix row
- d. Lookup table

- a.
 b.
 c.
 d.

No, the answer is incorrect.

Score: 0

Accepted Answers:

c.

8) Matrix-matrix multiplication using tiling can be helpful in CUDA optimization because 1 point

- a. Tiled part of the matrix can fit into shared memory
- b. Better communication to compute ratio
- c. More number of threads can be launched
- d. None of the above

- a.
 b.
 c.
 d.

No, the answer is incorrect.

Score: 0

Accepted Answers:

a.

b.

9) OpenACC cannot be used for parallelization of a code for the following platforms 1 point

- a. GPGPU
- b. Coprocessors
- c. Multicore symmetric multiprocessor
- d. Community cluster

- a.
 b.
 c.
 d.

No, the answer is incorrect.

Score: 0

Accepted Answers:

d.

10) In a hybrid multi-GPU program, MPI can be used along with CUDA for communication between the GPU-s 1 point

- a. True
- b. False

- a.
 b.

No, the answer is incorrect.

Score: 0

Accepted Answers:

a.