X


(https://swayam.gov.in)


(https://swayam.gov.in/nc_details/NPTEL)

Unit 11 - Week 8

Assignment 8

The due date for submitting this assignment has passed. Due on 2020-03-25, 23:59 IST.
As per our records you have not submitted this assignment.

Common data for question 1 to 4:

Consider the following two kernels executing on a GPU architecture where the number of SMs is 32 and the number of SPs per SM is 1024. For both kernels, the buffers out1, out2, in1 and in2 contain 4194304 elements. Let us consider fusing the two kernels using the different types of fusion methodologies taught. There are three different kernel launch parameters for each individual kernel listed below.

case 1: kernel1<<< 4096; 1024 >>>, kernel2<<< 4096; 1024 >>>

case 2: kernel1<<< 8192; 512 >>>, kernel2<<< 8192; 512 >>>

case 3: kernel1<<< 4096; 1024 >>>, kernel2<<< 4096; 1024 >>>

```
global void kernel1 (float  out1, float  in1, float  in2
)
{
     int  tid = …..
     out[tid] = in1[tid] + log( in2[tid] )
}
global void kernel2 (float  out2, float  in1, float  in2
)
{
     int  tid = …..
     out[tid] = sin(in1[tid]) + cos( in2[tid] )
}
```

Assume that each math operation (sin, cos, log) takes 6 cycles and each addition operation takes 4 cycles, and ignoring penalties incurred due to memory accesses and data transfer overhead.

1) ***10 points***

Assume you have used inner thread fusion for case1, inner block fusion for case 2 and inter block fusion for case 3. For each type of fusion methodology, write the corresponding kernel launch parameters for the fused kernel given the launch parameters for non fused kernels.

a. kernelF_innerthread<<< 4096; 1024 >>>, kernelF_innerblock<<< 8192; 1024 >>>, kernelF_interblock<<< 8192; 1024 >>>
b. kernelF_innerthread<<< 1024; 1024 >>>, kernelF_innerblock<<< 1024; 1024 >>>, kernelF_interblock<<< 1024; 1024 >>>
c. kernelF_innerthread<<< 2048; 1024 >>>, kernelF_innerblock<<< 2048; 1024 >>>, kernelF_interblock<<< 4096; 1024 >>>
d. kernelF_innerthread<<< 1024; 512 >>>, kernelF_innerblock<<< 4096; 1024 >>>, kernelF_interblock<<< 8192; 512 >>>

- a.
- b.
- c.
- d.

No, the answer is incorrect.
Score: 0

Accepted Answers:
a.

Details Solution

Download Videos

Text Transcripts

Live Interactive Session

2) ***15 points***

Calculate for Inner Thread Fusion, the total number of cycles it takes for the fused kernel to execute on the given architecture.

a. 3328
b. 4096
c. 1024
d. 3412

- a.
- b.
- c.
- d.

No, the answer is incorrect.
Score: 0

Accepted Answers:
a.

3) ***15 points***

Calculate for Inner Block Fusion, the total number of cycles it takes for the fused kernel to execute on the given architecture.

a. 3328
b. 4096
c. 1024
d. 3412

- a.
- b.
- c.
- d.

No, the answer is incorrect.
Score: 0

Accepted Answers:
b.

4) ***15 points***

Calculate for Inter Block Fusion, the total number of cycles it takes for the fused kernel to execute on the given architecture in the best case scenario.

a. 3328
b. 4096
c. 1024
d. 3412

- a.
- b.
- c.
- d.

No, the answer is incorrect.
Score: 0

Accepted Answers:
a.

5) ***30 points***

For the given GPU architecture-

- Streaming Multiprocessors (SM): 20
- Max. active threads per SM: 2048
- Max. thread blocks per SM: 32
- Registers per SM: 64K
- Max Shared Memory per SM: 96KB
- Max. Shared Memory per block: 48KB

The given kernel finds the maximum of a given data set (without any coarsening).

```
__global__ void max ( int * g_idata , int * g_odata ,
unsigned int n){
  __shared__ int sdata [1024];
  unsigned int tid = threadIdx .x;
  unsigned int i = blockIdx .x * ( blockDim .x * 2) +
threadIdx .x;
  __syncthreads ();
  for ( unsigned int s= blockDim .x /2; s>0; s > >=1) {
     if ( tid < s)
         sdata [ tid ] =max(sdata [ tid ], sdata [ tid
+ s]);
     __syncthreads ();
   }
   if ( tid == 0)
     g_odata [ blockIdx .x] = sdata [0];
}
```

Calculate the optimal coarsening factor for thread-level coarsening and block-level coarsening for the above program.

a. 4, 4
b. 8, 8
c. 12, 12
d. 16, 16

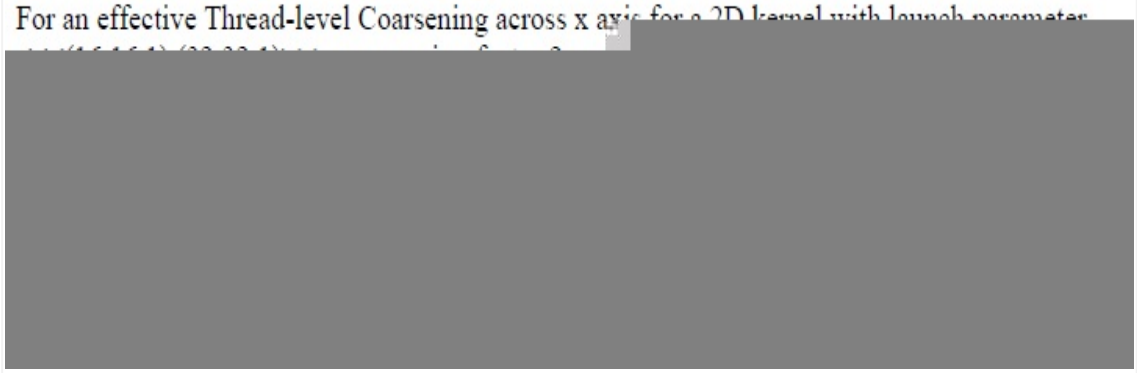
- a.
- b.
- c.
- d.

No, the answer is incorrect.
Score: 0

Accepted Answers:
b.

6) ***10 points***

For an effective Thread-level Coarsening across x axis for a 2D kernel with launch parameter



- a.
- b.
- c.
- d.

No, the answer is incorrect.
Score: 0

Accepted Answers:
b.

7) State if the following statement is true or false- ***5 points***

"Cache line re-use always improves performance"

a) True
b) False

- a.
- b.

No, the answer is incorrect.
Score: 0

Accepted Answers:
b.