NPTEL Online Certification Courses
Indian Institute of Technology Kharagpur

GPU Architecture and Programming

Assignment : Week 10

Type of Questions: Objective

Question 1

What is the advantage of implementing multiple command queues within the same context in a single device for an OpenCL code?

Options:

A. Helps in the faster data processing by improving synchronization
B. Reduces kernel launch overhead by having multiple command queues
C. Pipeline can be formed for better utilization by not having the device sit idle waiting for data
D. None of them

Answer: C

Question 2

Device fission can be helpful in following case(s)-

i) Dividing CPU-like devices into smaller sub-devices
ii) To build multiple OpenCL streams in GPUs
iii) To merge multiple devices to increase parallelism

Options:

A. i,ii,iii
B. i,ii
C. i
D. None of them

Answer: C

Question 3

Consider the reduction kernel to find the maximum of a given data set (65536 elements).

```
__global__ void max ( int * g_idata , int * g_odata , unsigned int n){
  __shared__ int sdata [2048];
  unsigned int tid = threadIdx .x;
  unsigned int i = blockIdx .x * ( blockDim .x * 2) + threadIdx .x;
  sdata [ tid ] = g_idata [i] + g_idata [i+ blockDim .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];
}
```

If you apply thread coarsening with coarsening factor 4 for a kernel. The maximum for a size of a work group that can be launched is 1024 work-items. The kernel invocation command is given below. What is the content of the arrays *global_work_size and local_work_size* ?

err = clEnqueueNDRangeKernel (commands, max_coarsened, 1, NULL, &global_work_size, &local_work_size , 0, NULL, NULL);

Options

A. global_work_size = {8192,1,1 }, local_work_size = {2048,1,1 }
B. global_work_size = {4194304,1,1} , local_work_size = {2048,1 ,1 }
C. global_work_size = {16384 , 1 , 1}, local_work_size = {1024 ,1,1 }
D. global_work_size = {131072,1 , 1}, local_work_size = { 1024, 1,1 } (correct)

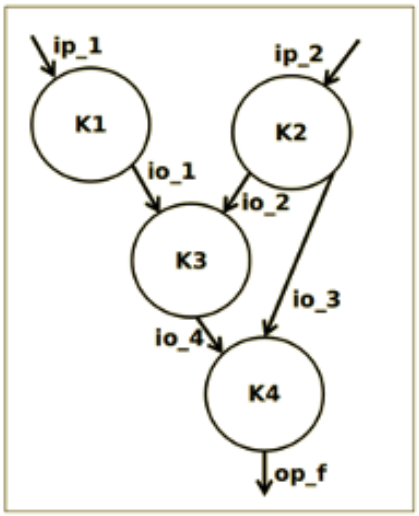
Answer: C

Solution: global_work_size = {2^16/2^2=2^14,1,1} ,

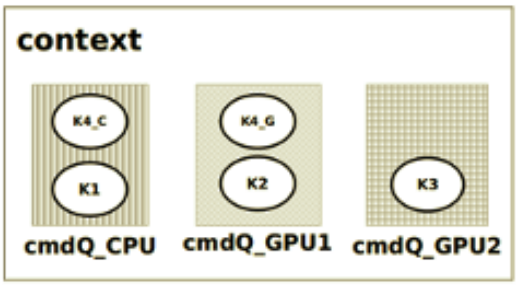
local_work_size = {1024,1 ,1 }

Common data for QUESTION 4-10:

Let us consider the Directed Acyclic Graph (DAG) given below, where each node denotes a kernel, and each edge denotes a dependency between two kernels. An edge from node **K1** to node **K3**, implies that the output of **K1** is used as input for **K3**. Thus, **K3** cannot start executing until both kernels **K1 and K2** have finished execution. The given DAG has a total of four kernels and is executed on a target platform that has one AMD CPU and two AMD GPUs. Each of these devices has a command queue associated with it. Kernels **K1, K2** and **K3** are assigned to CPU, GPU1 and GPU2 respectively. **K4** is partitioned in a ratio of 1:4 across CPU and GPU1 i.e. 1/(1+4) =20 % of the computation is mapped to the CPU and 4/(1+4) = 80 % of the computation is mapped to the GPU. The corresponding cl_kernel objects are `K1, K2, K3, K4_C` and `K4_G`. Let the variables `global` and `local` denote global_work_size and local_work_size respectively. The command queues related to the CPU, GPU1 and GPU2 are denoted by the variable names `cmdQ_CPU, cmdQ_GPU1` and `cmdQ_GPU2`.



Kernel DAG



Context and CommandQueues of target platform

We have an incomplete OpenCL code snippet designed for executing this DAG on the target platform.

Code snippet

```
cl_event event_cpu , event_gpu1, event_gpu2, events_1[2], events_2[2] ;
err = clEnqueueNDRangeKernel ( cmdQ_CPU, K1, 1, NULL, global, local, 0, NULL , &event_cpu );
events_1[0]=*(_____i_____);
err = clEnqueueNDRangeKernel ( cmdQ_GPU1, K2, 1, NULL, global, local, 0 , NULL , &event_gpu1 );
events_1[1]=*(_____ii_____);
err = clEnqueueNDRangeKernel ( cmdQ_GPU2, K3, 1, NULL, global,
local, ______iii______ ,______iv______ , &event_gpu2 );
events_2[0]=*(&event_gpu1);
events_2[1]=*(&event_gpu2);
err = clEnqueueNDRangeKernel ( cmdQ_CPU, K4_c, 1, NULL, global, local,  _____v_______
,______vi______ , &event_cpu);
err = clEnqueueNDRangeKernel ( cmdQ_GPU1, K4_g, 1, NULL, global, local, _______vii_____
,&events_2 , &event_gpu1 );
```

Question 4 :

Choose the correct option to fill in the blank in (i).

Options:

A. &event_gpu1
B. &event_cpu
C. &events_1
D. 1

Answer: B

Question 5 :

Choose the correct option to fill in the blank in (ii).

Options:

A. &event_gpu1
B. &event_gpu2
C. &events_1
D. &event_cpu

Answer: A

Question 6 :

Choose the correct option to fill in the blank in (iii).

Options:

A. 0
B. 1
C. 2
D. 3

Answer: C

Question 7 :

Choose the correct option to fill in the blank in (iv).

Options:

A. &event_gpu1
B. &event_gpu2
C. &events_1
D. &event_cpu

Answer: C

Question 8 :

Choose the correct option to fill in the blank in (v).

Options:

A. 0
B. 1
C. 2
D. 3

Answer: C

Question 9 :

Choose the correct option to fill in the blank in (vi).

Options:

A. &event_gpu1
B. &event_gpu2

C. &events_1
D. &events_2

Answer: D

Question 10 :

Choose the correct option to fill in the blank in (vii).

Options:

A. 0
B. 1
C. 2
D. 3

Answer: C