Test #2

- Due Jun 15 at 11:59pm
- Points 100
- Questions 40
- Available Jun 12 at 12:01am - Jun 15 at 11:59pm
- Time Limit 60 Minutes

Instructions

This is called a "quiz" on Canvas, but it is really 100-point Test #2.

There is a time limit of 60 minutes on this test. Once you start, you must finish. Canvas will not let you pause and come back later.

This test is Open Notes, Closed Friends, and Closed Internet.

This is your last graded-anything in this class, so do a good job!

Thanks for a *great* quarter!

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	42 minutes	100 out of 100

ⓘ Correct answers will be available on Jun 17 at 12:01am.

Score for this quiz: 100 out of 100
Submitted Jun 12 at 8:25am
This attempt took 42 minutes.

Question 1
2.5 / 2.5 pts

Why did Jane Parallel use those typedefs (point, vector, color, sphere) in her OpenCL particle system code?

○ Those were indeed the real OpenCL names for those types of variables

● It makes it more obvious what her code is doing

○ The compiler requires it

○ The OpenCL standard requires it

Question 2

2.5 / 2.5 pts

When OpenCL and OpenGL work together:

● They take turns accessing the vertex buffer

○ It is just OpenGL that is able to access the vertex buffer

○ They both access the vertex buffer at the same time

○ It is just OpenCL that is able to access the vertex buffer

Question 3

2.5 / 2.5 pts

The 2024 SC conference (International Conference for High Performance Computing, Networking, Storage, and Analysis) will be held:

● In Atlanta, GA

○ In Austin, TX

○ In Washington, DC

○ In Denver, CO

Question 4

2.5 / 2.5 pts

In MPI, a CPU’s “rank” is:

○ Its processing power

● Its integer identifier

○ Its priority

○ The number of cores it has

Question 5

2.5 / 2.5 pts

You intend to run your GPU program on the DGX server. While logged into one of the submit- machines, you need to type, for example:

○ ./montecarlo

● sbatch montecarlo.bash

○ ./montecarlo.cu

○ slurm montecarlo.bash

Question 6

2.5 / 2.5 pts

In Project #5, Joe Parallel put this in the C++ part of his .cu file:

```
int numSuccesses = 0;
for (int i = 0; i < NUMTRIALS; i++)
{
        numSuccesses += dsuccesses[i];
}
```

What can you say about this code?

○ It will work perfectly as-is

● It won't work because he is using a device array instead of a host array

○ It won't work because he is using a host array instead of a device array

○ It will work OK, but is unnecessary because CUDA could have completely done this automatically

Question 7

2.5 / 2.5 pts

The 2025 ACM SIGCSE conference will be held in:

○ Cambridge, MA

● Pittsburgh, PA

○ Denver, CO

○ Portland, OR

Question 8

2.5 / 2.5 pts

One of the ways that CUDA differs from OpenCL is:

○ CUDA GPU code looks like C and OpenCL GPU code looks like Python

● In CUDA, the C/C++ and GPU code are placed in the same file

○ CUDA GPU code looks like Python and OpenCL GPU code looks like C

○ In OpenCL, the C/C++ and GPU code are placed in the same file

Question 9

2.5 / 2.5 pts

Where is the OpenCL kernel compiler (as we used it this quarter) located?

○ On the Internet

● In the OpenCL driver

○ As an external program

○ In the GPU

Question 10

2.5 / 2.5 pts

A "CUDA Core":

○ Consists of multiple cores

○ Consists of flow control alone

● Consists of an integer unit and a floating-point unit

○ Consists of flow control, plus an integer unit and a floating-point unit

Question 11

2.5 / 2.5 pts

A Sphere can be represented as four floats. What are they?

● XYZ of the center position and the radius

○ The STP of the texture coordinates and the radius

○ The four hyperbolic radii

○ XYZ of the surface normal and the radius

Question 12

2.5 / 2.5 pts

MPI Reductions:

● Are a built-in feature of the MPI API

○ Are unnecessary because of the SIMD units on the CPUs

○ Must be implemented by your application

○ Are unnecessary because of the number of CPUs

Question 13

2.5 / 2.5 pts

What is one reason that OpenCL uses a Command Queue?

○ This paradigm is forced by how the hardware works

○ To be compatible with CPU-SIMD

● So OpenCL can gobble up commands as fast as it can

○ So that you don’t need to know what each command does

Question 14

2.5 / 2.5 pts

In this class, the letters “MPI” stand for:

● Message Passing Interface

○ MegaCalculation Per Instruction

○ Millions of Processor Instructions

○ Many-Processor Interfaces

Question 15

2.5 / 2.5 pts

What is the advantage of a Fused-Multiply-Add?

● It can perform a multiply plus an add in about the same time as it could have done the multiply alone

○ You only have to write one line of code instead of two

○ It implies that a SIMD operation should be performed

○ It reduces the possibility of False Sharing

Question 16

2.5 / 2.5 pts

What is special about using OpenCL/OpenGL interoperability?

○ The Khronos Group gives you a certificate for doing it

● The data never leaves GPU memory

○ It allows GPU graphics to be driven by CPU multicore computing

○ It saves electrical power

Question 17

2.5 / 2.5 pts

In MPI, a "communicator" is:

● The group of CPUs that have been allocated for this computation

○ The method used to perform a Scatter/Gather

○ The method used to communicate data values beween the CPUs

○ The method used to perform a Broadcast

Question 18

2.5 / 2.5 pts

Joe Parallel wants to use OpenCL kernels to implement the graph execution structure shown here. How?

I know that sometimes Canvas-quiz diagrams don't show up in certain browsers, so here is a textual description of the (perhaps missing) diagram:

It is a graph diagram like in the OpenCL Events notes with 4 kernels to execute: A, B, C, D. In the diagram:

- **A points to C and D**
- **B points to C**
- **C points to D**

● He has A, B, and C each throw events, and has C, C, and D (respectively) wait for those events.

○ He can't -- it is not possible in the current version of OpenCL, but might be in the future

○ He sets up barriers at C and D

○ He turns C and D into special OpenCL reduction functions

Question 19

2.5 / 2.5 pts

The OpenCL function clCreateFromGLBuffer( ):

○ Creates an OpenGL graphics vertex buffer object

● Creates an OpenCL device memory pointer from an OpenGL graphics vertex buffer object

○ Deletes an OpenCL buffer and replaces it with an OpenGL-compatible vertex buffer object

○ Allocates an OpenCL device memory buffer

Question 20

2.5 / 2.5 pts

Let's beat the Yellow Robot metaphor to death. The Yellow Robot's grippers represent:

○ Compute Units

○ Separate CPU cores

● Processing Elements

○ SIMD banks

Question 21

2.5 / 2.5 pts

In your CUDA program, how do you show that a function is a GPU Kernel function?

○ By labeling it with __device__

○ By labeling it with __kernel__

○ By labeling it with __local__

● By labeling it with __global__

Question 22

2.5 / 2.5 pts

In Project #3 (K-Means), how did we know if the k-means algorithm worked correctly?

○ The capital cities ended up not alphabetized even though they did start out that way

○ The capital cities ended up clustered all in one region

○ The capital cities ended up alphabetized even though they did not start out that way

● The capital cities ended up widely distributed

Question 23

2.5 / 2.5 pts

What is the relationship between Global Data Set Size, Work Group Size, and the Number of Work Groups?

○ Global Data Set Size = (Number of Work Groups)^2 [i.e., squared]

● Global Data Set Size = (Work Group Size) * (Number of Work Groups)

○ Work Group Size = (Global Data Set Size) * (Number of Work Groups)

○ Number of Work Groups = (Work Group Size) * (Global Data Set Size)

Question 24

2.5 / 2.5 pts

What does the OpenCL call “*gid = get_global_id(0)*” return?

○ It tells you how big the 1D global dataset is

○ It tells you how big the 1D local dataset is

● It tells you where you are in the 1D global dataset

○ It tells you where you are in the 1D local dataset

Question 25

2.5 / 2.5 pts

Function calls on GPU hardware:

● End up being inlined because there is no stack to store arguments and return addresses

○ Happen exactly the same way as CPU compiler/hardware implements them

○ Happen through the special “GPU-stack”, which is the same as a CPU-stack

○ Happen through the special “GPU-stack”, which is different from a CPU-stack

Question 26

2.5 / 2.5 pts

In in order to get an A grade in CS 475/575, you need to:

● Accumulate 1060 total points

○ Accumulate 95 total points

○ See a 97% on Canvas

○ See a 93% on Canvas

Question 27

2.5 / 2.5 pts

An MPI “Broadcast” operation involves:

○ Many function calls: a broadcast sender and one unique broadcast receiver function per CPU

○ Many function calls: multiple broadcast senders and a single broadcast receiver

● A single function call regardless of if you are sending or receiving

○ Two function calls: a broadcast sender and a broadcast receiver

Question 28

2.5 / 2.5 pts

In the CUDA call:

cudaMemcpy(A, B, NUM_ELEMENTS*sizeof(float), cudaMemcpyHostToDevice); **:**

○ The GPU array B gets copied to the CPU array A

○ The GPU array A gets copied to the CPU array B

● The CPU array B gets copied to the GPU array A

○ The CPU array A gets copied to the GPU array B

Question 29

2.5 / 2.5 pts

There were several cases when OpenCL, in querying what sort of system it was running on, called the same function twice:

```
status = clGetDeviceIDs( platform, CL_DEVICE_TYPE_ALL, 0, NULL,
&numDevices );
. . .
status = clGetDeviceIDs( platform, CL_DEVICE_TYPE_ALL, numDevices,
devices, NULL );
```

Why?

○ Once to get the information from a CPU/GPU, and once to get it from an FPGA (Field-Programmable Gate Array)

◉ Once to get the number of something, and once to retrieve that much information

○ So you could get the information from two separate platforms

○ Once to get the information from a CPU, and once to get it from a GPU

Question 30

2.5 / 2.5 pts

In a 1D problem, the Compute:Communicate ratio is (where N is the number of computations being done in each processor):

○ N : 4

◉ N : 2

○ N : 6

Question 31

2.5 / 2.5 pts

Comparing CPUs and GPUs, it is correct to say:

○ GPUs are better with linked-list data structures, CPUs are better with data parallel arrays

○ GPUs are better with integers, CPUs are better with floating-point

◉ CPUs are better with linked-list data structures, GPUs are better with data parallel arrays

○ CPUs are better with integers, GPUs are better with floating-point

Question 32

2.5 / 2.5 pts

Projects #1 and #5 ran roughly the same code on a CPU and a GPU, respectively. What can you say about their relative performance in Trials/

Second?

○ Within 10%, the two versions had about the same performance

● The GPU version was much faster

○ The CPU version was much faster

Question 33

2.5 / 2.5 pts

OpenCL Reductions:

● Must be implemented by the .cl function you write

○ Are unnecessary because of the GPU speed

○ Are a built-in feature of the OpenCL API just like they are in OpenMP

○ Are unnecessary because of the GPU hardware instruction set

Question 34

2.5 / 2.5 pts

In MPI, the phrase “scatter/gather” means:

○ To gather input data from the disk and then scatter it out to different CPUs for computation

● To break a problem up into pieces, give each piece to a separate CPU, and then gather up the results

○ To use the MPI_Bcast( ) function to get information to all other CPUs

○ To setup Barriers across all CPUs

Question 35

2.5 / 2.5 pts

The OpenGL-created Vertex Buffer Object looks, to OpenCL, like:

○ A hash table of XYZ arrays

○ A collection of separate X[ ], Y[ ], and Z[ ] arrays

○ A linked list of XYZ coordinates

● A table of XYZ coordinates

Question 36

2.5 / 2.5 pts

Like occurred in Project #7, what happens to your MPI executable when you run

it with *mpiexec*?

○ The program wakes up on the THEBOSS processor which then manually distributes it to the other processors

● The exact same executable wakes up at the same time on all processors

○ The program runs on one and only one processor total

Question 37

2.5 / 2.5 pts

What is SLURM used for?

○ It manages access to SIMD units

○ It manages access to a CPU's cores

○ It manages access to the fast network that connects machines in a cluster

● It manages access to shared computing resources such as the DGX and the cluster

Question 38

2.5 / 2.5 pts

Jane Parallel is combining multicore with SSE SIMD. If she is using N cores, what maximum speedup can she expect?

○ 4 + N

○ 2 * N

● 4 * N

○ 2 + N

Question 39

2.5 / 2.5 pts

In your C/C++ CUDA program, how do you show that you are making a call to the GPU Kernel function?

○ With the >>> … <<< (chevron) syntax

○ With the cudaEnqueueNDRangeKernel( ) function

○ With the cudaExecuteKernel( ) function

● With the <<< … >>> (chevron) syntax

Question 40

2.5 / 2.5 pts

What does the function cudaMalloc( ) do?

○ Allocates space in CPU memory

○ Pre-allocates space in the GPU cache

● Allocates space in GPU memory

○ Allocates space in both CPU and GPU memory

Quiz Score: 100 out of 100