Test #2

Due Jun 12 at 11:59pm **Points** 100 **Questions** 40
Available Jun 8 at 12:01am - Jun 12 at 11:59pm 5 days **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!

ⓘ Correct answers will be available on Jun 13 at 12:01am.

Score for this quiz: **100** out of 100
Submitted Jun 9 at 1:33pm
This attempt took 41 minutes.

Question 1 — 2.5 / 2.5 pts

What is one reason that OpenCL uses a Command Queue?

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

Question 2

2.5 / 2.5 pts

When OpenCL and OpenGL work together:

- (•) They take turns accessing the vertex buffer
- ( ) It is just OpenCL that is able to access the vertex buffer
- ( ) They both access the vertex buffer at the same time
- ( ) It is just OpenGL that is able to access the vertex buffer

Question 3

2.5 / 2.5 pts

In Project #3, the Functional Decomposition project, each individual quantity's function needed to have three barriers. The *second* barrier was there to:

- (•) Indicate when that quantity's next value was done being copied to the global state
- ( ) Indicate when that quantity's function was done computing that quantity's next value
- ( ) Indicate when it was time to increment the month
- ( ) Indicate when the Watcher thread could print values

Question 4

2.5 / 2.5 pts

The advantage of using SSE SIMD is:

- [x] You should be able to get a 4x performance improvement in array multiplication
- [ ] It uses a stack variable to hold each multiplication product
- [ ] You can perform a multiply and an add in one instruction
- [ ] It uses the stack to hold its pointers

Question 5

2.5 / 2.5 pts

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

- [ ] Millions of Processor Instructions
- [x] Message Passing Interface
- [ ] MegaCalculation Per Instruction
- [ ] Many-Processor Interfaces

Question 6

2.5 / 2.5 pts

In MPI, a "derived type" is:

- [ ] Being able to pack multiple MPI_INTs into a single MPI_LONG
- [x] You creating a struct that can act just like MPI_FLOAT, MPI_INT, etc.
- [ ] Being able to pack multiple MPI_CHARs into a single MPI_LONG

Linking multiple MPI calls together to send both MPI_FLOATs and MPI_INTs

Question 7

2.5 / 2.5 pts

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

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

He turns C and D into special OpenCL reduction functions

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

He sets up barriers at C and D

Question 8

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 cudaExecuteKernel( ) function

- (selected) With the <<< … >>> (chevron) syntax
- With the cudaEnqueueNDRangeKernel( ) function
- With the >>> … <<< (chevron) syntax

Question 9

2.5 / 2.5 pts

In Project #3, the Functional Decomposition project, each individual quantity's function needed to have three barriers. The *first* barrier was there to:

- Indicate when that quantity's next value was done being copied to the global state
- Indicate when the Watcher thread could print values
- Indicate when it was time to increment the month
- (selected) Indicate when that quantity's function was done computing that quantity's next value

Question 10

2.5 / 2.5 pts

MPI Reductions:

- Must be implemented by your application
- Are unnecessary because of the SIMD units on the CPUs
- Are unnecessary because of the number of CPUs

- [x] Are a built-in feature of the MPI API

Question 11

2.5 / 2.5 pts

When is it OK to use the less-precise “fast_normalize( )” call instead of the full-precision “normalize( )” call?

- [ ] Always
- [ ] When using OpenCL for scientific computing
- [x] When using OpenCL for computer graphics
- [ ] Never

Question 12

2.5 / 2.5 pts

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

- [ ] The compiler requires it
- [x] It makes it more obvious what her code is doing
- [ ] The OpenCL standard requires it
- [ ] Those were indeed the real OpenCL names for those types of variables

Question 13

2.5 / 2.5 pts

As of the writing of our class notes, the 2022 IEEE Visualization conference:

- Will be online-only
- Has been cancelled
- Will be held in Cambridge, MA
- [x] Will be held in Oklahoma City, OK

Question 14

2.5 / 2.5 pts

What does the function cudaMalloc( ) do?

- Allocates space in CPU memory
- Pre-allocates space in the GPU cache
- [x] Allocates space in GPU memory
- Allocates space in both CPU and GPU memory

Question 15

2.5 / 2.5 pts

One of the ways that CUDA differs from OpenCL is:

- CUDA GPU code looks like Python and OpenCL GPU code looks like C
- [x] In CUDA, the C/C++ and GPU code are placed in the same file

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

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

Question 16

2.5 / 2.5 pts

MPI follows what parallel programming model?

- (●) Single Program, Multiple Data (SPMD)
- ( ) Single Instruction, Single Data (SISD)
- ( ) Multiple Instructions, Multiple Data (MIMD)
- ( ) Single Instruction, Multiple Data (SIMD)

Question 17

2.5 / 2.5 pts

Function calls on GPU hardware:

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

Happen exactly the same way as CPU hardware implements them

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

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

Question 18

2.5 / 2.5 pts

GPU Reductions:

- (selected) Must be implemented by the .cl function you write
- Are unnecessary because of the GPU hardware instruction set
- Are unnecessary because of the GPU speed
- Are a built-in feature of the OpenCL API just like they are in OpenMP

Question 19

2.5 / 2.5 pts

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

- By labeling it with __device__
- (selected) By labeling it with __global__
- By labeling it with __local__
- By labeling it with __kernel__

Question 20

2.5 / 2.5 pts

In CUDA, how many threads are in each Warp?

- 128
- 16
- 64
- (•) 32

Question 21

2.5 / 2.5 pts

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

- SIMD banks
- (•) Processing Elements
- Compute Units
- Separate CPU cores

Question 22

2.5 / 2.5 pts

A "CUDA Core":

- Consists of multiple cores
- (•) Consists of an integer unit and a floating-point unit
- Consists of flow control, plus an integer unit and a floating-point unit
- Consists of flow control alone

Question 23

2.5 / 2.5 pts

Let's beat the Yellow Robot metaphor to death. The yellow robot represents:

- ○ A Processing Element
- ○ A separate CPU core
- ○ A SIMD bank
- ◉ A Compute Unit

Question 24

2.5 / 2.5 pts

In running your GPU program on the DGX server, you needed to type, for example:

- ○ slurm montecarlo.csh
- ○ ./montecarlo
- ○ ./montecarlo.cu
- ◉ sbatch montecarlo.bash

Question 25

2.5 / 2.5 pts

In the OpenCL call "gid = get_global_id(0)", what does the argument of 0 indicate?

That you want only one value returned

Since the time at which the program started

In the X dimension

Relative to the first element of the dataset

Question 26

2.5 / 2.5 pts

In Project #1, you performed a multicore Monte Carlo simulation by using the NUMTRIALS for-loop. In Project #5, you re-created that same simulation using CUDA without any for-loop. Where did that NUMTRIALS for-loop go?

It is still there – it has just been written in CUDA-code instead of C/C++

It is still there – it has just been replaced with the special CUDA "foreach" capability

You don't need to include it– CUDA is smart enough to figure out what you are trying to do and adds it for you

It is not needed – it has been replaced by duplicating the simulation onto thousands of threads

Question 27

2.5 / 2.5 pts

What is special about using OpenCL/OpenGL interoperability?

- It saves electrical power
- The Khronos Group gives you a certificate for doing it
- (selected) The data never leaves GPU memory
- It allows GPU graphics to be driven by CPU multicore computing

Question 28

2.5 / 2.5 pts

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

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

Question 29

2.5 / 2.5 pts

In Project #3, the Functional Decomposition project, each individual quantity's function needed to have three barriers. The *third* barrier was there to:

- Indicate when that quantity's next value was done being copied to the global state

Indicate when that quantity's function was done computing that quantity's next value

Indicate when it was time to increment the time of day

Indicate when the Watcher thread was done printing values

Question 30

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, and once to get it from a GPU

So you could get the information from two separate platforms

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

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

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

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

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

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

Question 32

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 four hyperbolic radii

The STP of the texture coordinates and the radius

XYZ of the surface normal and the radius

Question 33

2.5 / 2.5 pts

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

It reduces the possibility of False Sharing

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

Question 34

2.5 / 2.5 pts

In the CUDA call:

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

- (selected) The CPU array B gets copied to the GPU array A
- The CPU array A gets copied to the GPU array B
- The GPU array A gets copied to the CPU array B
- The GPU array B gets copied to the CPU array A

Question 35

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?

- (selected) The GPU version was way faster
- The CPU version was way faster
- Within 10%, the two versions had about the same performance

Question 36

2.5 / 2.5 pts

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

- ○ A collection of separate X[ ], Y[ ], and Z[ ] arrays
- ○ A hash table of XYZ arrays
- ○ A linked list of XYZ coordinates
- ◉ A table of XYZ coordinates

Question 37

2.5 / 2.5 pts

The primary purpose of MPI is to:

- ◉ Allow parallel computing among separate computers
- ○ To allow multicore
- ○ To get computing access to a GPU
- ○ To get SIMD performance

Question 38

2.5 / 2.5 pts

What does the OpenCL call "gid = get_global_id(0)" return?

- ○ It tells you where you are in the local dataset
- ○ It tells you how big the global dataset is
- ○ It tells you how big the local dataset is
- ◉ It tells you where you are in the global dataset

Question 39

2.5 / 2.5 pts

Jane Parallel uses this line of OpenCL code:

status = clEnqueueNDRangeKernel(cmdQueue, kernel, 1, NULL, A, B, C, D, E);

what are the C and D variables used for?

- They specify what event to throw when this kernel is completed
- The globalWorkSize and the localWorkSize
- They specify how many events to wait for and which ones they are
- The context to use

Question 40

2.5 / 2.5 pts

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

- As an external program
- In the OpenCL driver
- On the Internet
- In the GPU

Quiz Score: **100** out of 100