Test #2

截止日期 6月12日 23:59 **分數** 100 **問題** 40
可用 6月8日 0:01 - 6月12日 23:59 5 天 **時間限制** 60 分鐘

說明

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!

▶ 試記錄

	嘗試	時間	分數
最新的	嘗試 1	43 分鐘	得分：100；總分：100

ⓘ 正確答案將於 6月13日 0:01 可用。

此測驗的分數： 得分：**100**；總分：100
已提交6月9日 19:12
此嘗試持續 43 分鐘。

問題 1 **2.5 / 2.5 分數**

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

- (●) XYZ of the center position and the radius
- ( ) The four hyperbolic radii
- ( ) XYZ of the surface normal and the radius
- ( ) The STP of the texture coordinates and the radius

問題 2

2.5 / 2.5 分數

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

問題 3

2.5 / 2.5 分數

The primary purpose of MPI is to:

- ( ) To get computing access to a GPU

- To get SIMD performance
- To allow multicore
- (•) Allow parallel computing among separate computers

問題 4

2.5 / 2.5 分數

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

問題 5

2.5 / 2.5 分數

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

- (•) To break a problem up into pieces, give each piece to a separate computers, and then gather up the results
- To setup Barriers across all CPUs
- To gather input data from the disk and then scatter it out to different CPUs for computation

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

問題 6

2.5 / 2.5 分數

MPI follows what parallel programming model?

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

問題 7

2.5 / 2.5 分數

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

- Compute Units
- Processing Elements
- Separate CPU cores
- SIMD banks

問題 8

2.5 / 2.5 分數

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

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

問題 9

2.5 / 2.5 分數

GPU Reductions:

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

問題 10

2.5 / 2.5 分數

A "CUDA Core":

- Consists of multiple cores
- [x] 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

問題 11

2.5 / 2.5 分數

What does the function cudaMalloc( ) do?

- Allocates space in both CPU and GPU memory
- Allocates space in CPU memory
- Pre-allocates space in the GPU cache
- (●) Allocates space in GPU memory

問題 12

2.5 / 2.5 分數

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?

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

問題 13

2.5 / 2.5 分數

What is special about using OpenCL/OpenGL interoperability?

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

問題 14

2.5 / 2.5 分數

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 the Watcher thread was done printing values
- Indicate when it was time to increment the time of day
- Indicate when that quantity's function was done computing that quantity's next value
- Indicate when that quantity's next value was done being copied to the global state

問題 15

2.5 / 2.5 分數

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

- ○ Always
- ○ Never
- ◉ When using OpenCL for computer graphics
- ○ When using OpenCL for scientific computing

問題 16

2.5 / 2.5 分數

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

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

問題 17

2.5 / 2.5 分數

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 sets up barriers at C and D
- He has A, B, and C each throw events, and has C, C, and D (respectively) wait for those events.
- He turns C and D into special OpenCL reduction functions

問題 18

2.5 / 2.5 分數

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

問題 19

2.5 / 2.5 分數

In MPI, a "derived type" is:

- Being able to pack multiple MPI_CHARs into a single MPI_LONG
- Being able to pack multiple MPI_INTs into a single MPI_LONG
- Linking multiple MPI calls together to send both MPI_FLOATs and MPI_INTs
- **You creating a struct that can act just like MPI_FLOAT, MPI_INT, etc.**

問題 20

2.5 / 2.5 分數

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

問題 21

2.5 / 2.5 分數

The advantage of using SSE SIMD is:

- You should be able to get a 4x performance improvement in array multiplication

- It uses the stack to hold its pointers
- You can perform a multiply and an add in one instruction
- It uses a stack variable to hold each multiplication product

問題 22

2.5 / 2.5 分數

In CUDA, how many threads are in each Warp?

- (selected) 32
- 64
- 16
- 128

問題 23

2.5 / 2.5 分數

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

- (selected) In the X dimension
- That you want only one value returned
- Since the time at which the program started
- Relative to the first element of the dataset

問題 24

2.5 / 2.5 分數

As of the writing of our class notes, the 2022 SC conference (International Conference for High Performance Computing, Networking, Storage, and Analysis) will be held:

- In Washington, DC
- Totally online
- In Los Angeles, CA
- In Dallas, TX

問題 25

2.5 / 2.5 分數

One of the ways that CUDA differs from OpenCL is:

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

問題 26

2.5 / 2.5 分數

When OpenCL and OpenGL work together:

- 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
- (●) They take turns accessing the vertex buffer

問題 27

2.5 / 2.5 分數

MPI Reductions:

- Must be implemented by your application
- Are unnecessary because of the number of CPUs
- (●) Are a built-in feature of the MPI API
- Are unnecessary because of the SIMD units on the CPUs

問題 28

2.5 / 2.5 分數

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

- slurm montecarlo.csh
- (●) sbatch montecarlo.bash

- ./montecarlo
- ./montecarlo.cu

問題 29

2.5 / 2.5 分數

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

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

問題 30

2.5 / 2.5 分數

The OpenCL function clCreateFromGLBuffer( ):

- Allocates an OpenCL device memory buffer
- Creates an OpenGL graphics vertex buffer object
- Deletes an OpenCL buffer and replaces it with an OpenGL-compatible vertex buffer object

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

問題 31

2.5 / 2.5 分數

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

問題 32

2.5 / 2.5 分數

An MPI “Broadcast” operation involves:

Many functions: multiple broadcast senders and a single broadcast receiver

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

Two functions: a broadcast sender and a broadcast receiver

A single function regardless of if you are sending or receiving

問題 33

2.5 / 2.5 分數

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 it was time to increment the month

Indicate when the Watcher thread could print values

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

問題 34

2.5 / 2.5 分數

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

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

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
- It implies that a SIMD operation should be performed

問題 35

2.5 / 2.5 分數

Jane Parallel uses this line of OpenCL code:

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

what is the E variable used for?

- The globalWorkSize
- It specifies what event to throw when this kernel is completed
- It specifies how many events to wait for
- The context to use

問題 36

2.5 / 2.5 分數

In the CUDA call:

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

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

- 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 A gets copied to the GPU array B

問題 37

2.5 / 2.5 分數

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

- (●) It tells you where you are in the global dataset
- It tells you how big the local dataset is
- It tells you where you are in the local dataset
- It tells you how big the global dataset is

問題 38

2.5 / 2.5 分數

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

- A separate CPU core
- A SIMD bank
- A Processing Element
- (●) A Compute Unit

問題 39

2.5 / 2.5 分數

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

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

問題 40

2.5 / 2.5 分數

Function calls on GPU hardware:

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

測驗分數： 得分：**100**；總分：100