Test #1

- Due May 5 at 11:59pm
- Points 100
- Questions 40
- Available May 1 at 12:01am - May 5 at 11:59pm
- Time Limit 60 Minutes

Instructions

Canvas calls this a "Quiz", but it is really Test #1.

It consists of 40 multiple choice questions to be done in 60 minutes. It is Open Notes.

Once you start, you must finish. Canvas will not let you pause and come back.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	58 minutes	97.5 out of 100

Correct answers will be available on May 6 at 12:01am.

Score for this quiz: 97.5 out of 100

Submitted May 4 at 2:27pm

This attempt took 58 minutes.

Question 1

2.5 / 2.5 pts

False Sharing happens because:

● One core is writing to a cache line at the same time another core is reading or writing the same cache line

○ Two cores are reading from the same cache line

○ One core is writing to a cache line at the same time another core is reading or writing a different cache line

○ Two cores are not sharing the same cache line, but should be

Question 2

2.5 / 2.5 pts

The cache that is smallest and fastest is named:

○ L2

○ L0

○ L3

● L1

IncorrectQuestion 3

0 / 2.5 pts

The reason that our OpenMP programs have a NUMTRIES for-loop is to:

○ Determine the peak performance

● Determine the median performance

○ Determine the standard deviation of performance

○ Determine the range of performance numbers

Question 4

2.5 / 2.5 pts

The word “deterministic” means:

○ The program outputs change every time you run the program

○ It describes a quantity that you are attempting to determine

● The same inputs will always produce the same outputs

○ The program outputs change whenever you change the number of threads

Question 5

2.5 / 2.5 pts

To get an A in CS 475/575 requires:

○ A weighted average of 93%

● 1060 total points

○ A weighted average of 96%

Question 6

2.5 / 2.5 pts

The difference between static and dynamic scheduling of an OpenMP for-loop is:

○ Dynamic scheduling divides all the for-loop passes among the threads at first

○ Dynamic scheduling changes the chunksize while the for-loop is running

● Dynamic scheduling divides only some of the for-loop passes among the threads at first

○ Dynamic scheduling allows you to change how the for-loop passes are divided up while they are running

Question 7

2.5 / 2.5 pts

Coarse-grained parallelism is:

○ Dividing the problem into pieces, of all which have to be a different size

● Dividing the problem into a small number of large pieces

○ Dividing the problem into equal-size pieces

○ Dividing the problem into a large number of small pieces

Question 8

2.5 / 2.5 pts

In an n-core multicore program, what do you need to do to compute the $F_{parallel}$?

○ Go find out the size of the cache and use the inverse Amdahl's Law

○ Measure just the 20-core performance and use the inverse Amdahl's Law

● Measure the Speedup and use our inverse Amdahl's Law

○ Figure out how many CPU sockets are in use and use the inverse Amdahl's Law

Question 9

2.5 / 2.5 pts

A Monte Carlo probability is computed by:

○ Subtracting the number of successes from the number of trials

○ Dividing the number of trials by the number of successes

● Dividing the number of successes by the number of trials

○ Adding the number of successes to the number of trials

Question 10

2.5 / 2.5 pts

Intel recently broke the CPU clock speed record by:

○ Running the CPU outside during a colder-than-usual winter

○ Cooling the chip with liquid FlourInert

○ Cooling the chip with four fans

● Cooling the chip with liquid helium

Question 11

2.5 / 2.5 pts

A thread’s state consists of:

○ Stack, Program counter, Registers

○ Stack pointer, Stack, Registers

● Stack pointer, Program counter, Registers

○ Stack pointer, Program counter, Stack

Question 12

2.5 / 2.5 pts

A good way to make a piece of code *not* Thread Safe is to:

○ Use a mutual exclusion lock

○ Use a chunksize of 1

● Keep internal state

○ Use a private variable

Question 13

2.5 / 2.5 pts

In CS 475/575, the maximum number of Bonus Days that you can use on any one projects is:

● 2

○ 3

○ 4

○ 5

Question 14

2.5 / 2.5 pts

Moore’s Law (as Gordon Moore *actually* phrased it) says:

● Transistor density doubles every 2 years

○ The number of cores doubles every 2 years

○ Parallel fraction doubles every 2 years

○ Clock speed doubles every 2 years

Question 15

2.5 / 2.5 pts

When adding up the elements of a 2D array in C or C++, it is faster to add the elements:

○ Vetically (i.e., down the columns) first

● Horizontally (i.e., across the rows) first

○ It makes no speed difference either way

Question 16

2.5 / 2.5 pts

SPMD stands for:

○ Single Program, Much Data

○ Significant Parallelism, Much Data

○ Significant Parallelism, Multiple Data

● Single Program, Multiple Data

Question 17

2.5 / 2.5 pts

The two types of coherence that caches want to see in order to deliver maximum performance are:

● Spatial and Temporal

○ Systemic and Thermal

○ Spatial and Thermal

○ Systemic and Temporal

Question 18

2.5 / 2.5 pts

Which of these is an example of a forbidden *inter-loop dependency*?

○ a[i] = (float)(i);

● a[i] = a[i-1] + 1.;

○ a[i] = b[i] + 1.;

○ a[i] = 2.*a[i];

Question 19

2.5 / 2.5 pts

The difference between using OpenMP Tasks vs. using OpenMP Sections is that:

○ Tasks are statically allocated, sections are dynamic

○ Sections are deprecated

● Tasks are dynamically allocated, sections are static

○ Nothing -- they are different words for the same thing

Question 20

2.5 / 2.5 pts

The theoretical maximum speedup that you can *ever* achieve, no matter how many cores you add, is:

● 1/(1-Fp)

○ 1/Fp

○ Fs

○ 1/(Fp+Fs)

Question 21

2.5 / 2.5 pts

What does this code cause to happen?

```
#pragma omp atomic
      sum = sum + partialSum;
```

● Guarantees that the entire statement happens with no chance of interruption

○ Imitates the same functionality as an OpenMP collapse clause

○ Automatically makes the variable sum a private variable

○ Automatically makes the variable sum a shared variable

Question 22

2.5 / 2.5 pts

A “race condition” is one where:

○ It matters which thread gets to a barrier first

● You get a different result depending on which thread gets to a piece of code first

○ You get the same result regardless of which thread gets to a piece of code first

○ It matters which stack holds a particular variable

Question 23

2.5 / 2.5 pts

Gustafson’s Observation on Amdahl’s Law says:

● When people buy more cores they often do it to process more data, which results in a larger parallel fraction

○ When people buy more cores they often do it to reduce memory contention, which decreases performance

○ Amdahl’s Law only applies when you have a number of cores that is less than or equal to 8

○ Amdahl’s law was applicable when it was formulated, but doesn’t apply now

Question 24

2.5 / 2.5 pts

MESI stands for:

○ Nothing -- it's someone's name

○ Modified-Exclusive-Single-Invalid

○ Multicore-Exclusive-Shared-Invalid

○ Modified-Exclusive-Shared-Instructions

● Modified-Exclusive-Shared-Invalid

○ Modified-Exterior-Shared-Invalid

Question 25

2.5 / 2.5 pts

The advantage of using the OpenMP *reduction* clause is

● It greatly speeds, and makes thread-safe, reduction operations

○ No advantage, it is just cleaner code

○ It is less likely to result in a compiler error

○ Actually a disadvantage -- it can produce wrong, non-deterministic answers

Question 26

2.5 / 2.5 pts

Speedup Efficiency is defined as:

○ Fp/n

○ n

● Sn/n

○ Fp

Question 27

2.5 / 2.5 pts

OpenMP Reductions are faster than Atomic or Critical because:

○ They sum into an array whose elements are a Fibonacci series in size

○ They momentarily disable interrupts to keep the summing equation from being corrupted

● They sum into a separate variable per thread and then perform power-of-two addition

○ They sum into a user-supplied array and then let the programmer decide how to best sum them

Question 28

2.5 / 2.5 pts

A Barrier is:

○ A place in the code that threads are not allowed to pass ever

○ A place in the code where threads can spawn other threads

● A place in the code that all threads must reach before any of them are allowed to continue

○ A place in the code where the first thread to get there issues an interrupt

Question 29

2.5 / 2.5 pts

A Deadlock condition is when:

○ When it is a race to see which of two threads get to a piece of code first

○ When you keep internal state

○ The CPU chip cannot find any more instructions to execute while waiting for a memory fetch

● Two threads are each waiting for the other one to do something

Question 30

2.5 / 2.5 pts

One way to prevent harm from race conditions is:

○ Dynamic scheduling

● Mutual Exclusion Locks

○ Shared variables

○ Private variables

Question 31

2.5 / 2.5 pts

The cache that is closest to the Arithmetic Logic Unit (ALU) is named:

○ L0

○ L3

● L1

○ L2

Question 32

2.5 / 2.5 pts

Our class's "Inverse Amdahl's Law" that you used in Projects #0 and #1 computes:

○ Thread Efficiency, given Sn and n

● Fp, given Sn and n

○ Sn, given Fp and n

○ n, given Sn and Fp

Question 33

2.5 / 2.5 pts

The purpose of the Watcher Thread in our Functional Decomposition example program is to:

○ Time the simulation

○ Figure out what the animal or plant threads need to do next

● Print results, update the current month/year, and update environmental variables

○ Draw a picture of what is going on in the simulation

Question 34

2.5 / 2.5 pts

Why is there a photo of a carton of eggs in the Cache notes?

○ It explains Temporary Coherence

● Bringing home a dozen eggs when you only need 2 today is like the way cache works

○ Because the size of a cache line is a dozen floats

○ It explains Stationary Coherence

Question 35

2.5 / 2.5 pts

A "Mutex" is:

○ A sound you make when you sneeze

○ A "multiple texture" for graphics processing

○ A "mutual text" message

● Another term for a "mutual exclusion lock"

Question 36

2.5 / 2.5 pts

Hyperthreading is:

○ Adding more memory bandwidth

○ Adding one or more cores

● Keeping one or more extra thread states within a core

○ Adding extra cache space

Question 37

2.5 / 2.5 pts

In multithreading, the threads all share:

○ Heap, Execution instructions, and the same Stack

○ Execution instructions, Global variables, and the same Stack

○ Heap, Global variables, and the same Stack

● Heap, Execution instructions, and Global variables

Question 38

2.5 / 2.5 pts

Using “default(none)” in an OpenMP #pragma is:

○ A deprecated feature of an older version of OpenMP

● A good idea, but not required

○ A way to possibly increase performance

○ Required

Question 39

2.5 / 2.5 pts

The OpenMP *collapse* clause is used to:

○ Turn cascading if-statements into a single compound if-statement

● Allow the parallelization of more than one nested for-loop

○ Turn a group of constants multiplied together into a single constant

○ Unroll a for-loop

Question 40

2.5 / 2.5 pts

The *observation* that *clock speed* doubles every 2 years:

● Was the case for a while, but does not apply anymore

○ Has been correct starting in 1965 and is still happening

○ Is only correct for CPUs, not GPUs

○ Was never actually observed on real systems

Quiz Score: 97.5 out of 100