Contents

Lecture 2

Question 1

Today, what are the fundamental limits to the execution speed of a single CPU?

Question 2

Explain how caches can reduce the average memory access time.

Question 3

How can instruction level parallelism improve execution speed? What are the main approaches for implementing instruction level parallelism?

Question 4

Give one example of each type of machines in Flynn's taxonomy.

Lecture 3

Question 1

[Repetition from 62588] What is a process? What is a thread? What are the main differences and similarities between the two.

Question 2

[Repetition from 62588] What is a race condition? How can you remove a race condition from your program?

Question 3

What is the speedup of a parallel program?

Question 4

What is the main message of Amdahl's law?

Lecture 4

Question 1

Why do we need to start an MPI program with MPI_Init and end it with MPI_Finalize?

Question 2

In MPI, what is a communicator?

Lecture 5

Question 1

What is the point of collective communication in MPI? Can't all possible communication patterns be implemented with point-to-point communication primitives?

Question 2

In MPI, there is an MPI_Barrier primitive? What does it do? Isn't it odd for MPI to have such a primitive? In what situations is it useful?

Lecture 7

Question 1

What does it mean to parallelize code with OpenMP?

Question 2

Why are #pragma directives needed in OpenMP?

Lecture 8

read chapters 5.1 - 5.8

Question 1

Why can you not break out of a parallel for loop in OpenMP using return or a goto?

Question 2

Why are there different types of OpenMP scheduling?

Question 3

How do you define a critical section in OpenMP?