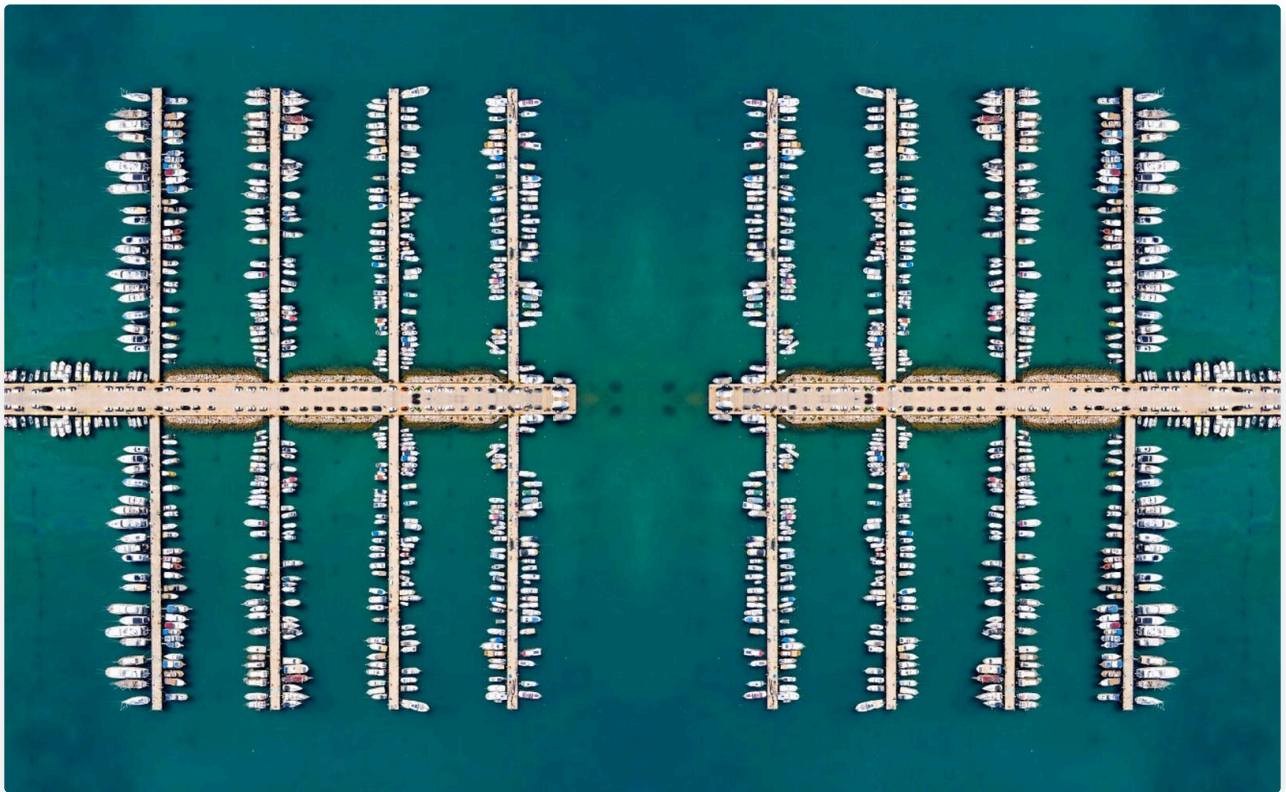


Sequential and Parallel Computing

Textbook

Sequential and Parallel Computing



Sequential Computing

[Sequential computing](#) is a computational model in which operations are performed in order one at a time. Any written program is just a series of instructions for a computer to follow. It's kind of like a recipe. If you follow the instructions on the recipe, you will come out with a meal. If a computer follows the instructions on a page of code, it will come out with a program.

New programmers are often taught to create sequential programs because they are easy to understand. All of the code you will learn in this course is sequential. One command happens after another.

Let's look at sequential programming through the example of baking cookies.

When making cookies, each ingredient needs to be carefully measured out before putting it in the mixing bowl.

1 min Flour

1 min Baking Soda

1 min Eggs

1 min Sugar

1 min Butter

1 min Vanilla

Total time to make cookie dough: **6 minutes**

If you were to make cookies sequentially, you would measure out each ingredient, one at a time, and put it in the mixing bowl. Then you would put the cookies into the oven. Let's say each step took you 1 minute to complete. Since there are 6 ingredients, it would take you 6 minutes to make the cookie dough.

A sequential solution takes as long as the sum of all of its steps.

Parallel Computing

[Parallel computing](#) is a computational model where the program is broken into multiple smaller sequential computing operations, some of which are performed simultaneously.

Let's look at the cookie example again.

With [sequential computing](#), each ingredient is measured out one at a time. With parallel computing, let's say we have another person in the kitchen to help you. You can both measure out ingredients at the same time. This way, both of you would measure out 3 ingredients, and then combine the ingredients together, so it would take a total of 4 minutes to make the cookie dough.

Person 1	Person 2
1 min Flour	1 min Sugar
1 min Baking Soda	1 min Butter
1 min Eggs	1 min Vanilla
1 min Combine Ingredients	

Total time to make cookie dough: 4 minutes

[Parallel computing](#) is much faster, but also more complex. Some bits of code are dependent on the completion of other bits of code before they can run. For example, if you were measuring out the cups of flour, the other person would need to wait for you to finish so they could use the measuring cup as well. This means that just because you have 2 people in the kitchen, that doesn't necessarily mean it will go twice as fast. You need to account for time spent communicating between the two people in order for the finished product to come together correctly.

Parallel computing is much more complex to create but it makes it so programs can run faster and you can compute more things in a shorter time frame. Often, when programs are analyzed, we compare the amount of time it takes for the code to run.

Solutions that use parallel computing can scale more effectively than solutions that use sequential computing.

Parallel Solutions Use Sequential Parts

Even [parallel computing](#) has sequential parts to the code. For example, with one person in the kitchen, after the ingredients have been measured, the dough needs to be spooned out on a tray and put in the oven. With two people in the kitchen, after they both measure the ingredients, the dough still needs to be spooned out on a tray and put in the oven. Certain steps still need to happen one after the other, even in parallel computing.

Speed

Comparing efficiency of solutions can be done by comparing the time it takes them to perform the same task.



Timing Cookie Sequential Example

A sequential solution takes as long as the sum of all of its steps. Let's say spooning out the dough on a tray takes 3 minutes and putting it in the oven takes 1 minute. Then you need to wait 10 minutes. So if you were alone in the kitchen it would take you 20 minutes to make the cookies.

1 min Flour

1 min Baking Soda

1 min Eggs

1 min Sugar

1 min Butter

1 min Vanilla

1 min Combine Ingredients

3 min Spoon onto Tray

10 min Baking Time

Total time to make cookies: 20 minutes

Parallel Cookie Example Timing

A parallel computing solution takes as long as its sequential tasks plus the longest of its parallel tasks. Let's see what that looks like in the cookie example.

Total time to make cookies: 17 minutes

So parallel systems complete programs faster, but do have their limitations.

A sequential solution takes as long as the sum of all of its steps.

A parallel computing solution takes as long as its sequential tasks plus the longest of its parallel tasks.

Speed Up

The "[speedup](#)" of a parallel solution is measured in the time it took to complete the task sequentially divided by the time it took to complete the task when done in parallel.

In the cookie example, it took 20 minutes sequentially and 17 minutes in parallel. So the speedup is $20/17$, or 1.176.

For another example, if it took 40 minutes sequentially and 25 minutes in parallel, what would the speedup be?

Answer: 1.6

When increasing the use of parallel computing in a solution, the efficiency of the solution is still limited by the sequential portion. This means that at some point, adding parallel portions will no longer meaningfully increase efficiency.

Distributed Computing



A program that runs with [distributed computing](#) leverages many computers or even computer networks to run a program.

Let's look at the cookie example again.

What if you just got a cookie mix? So the mix was prepared in a factory and packaged and sent to you? So when you got the mix, you only needed to spoon it onto the tray and put it in the oven? This would go much faster because the factory is very efficient at preparing mixes, which saves you time.

3 min Spoon onto Tray

10 min Baking Time

Total time to make cookies: 13 minutes

Distributed computing works in a similar way. Different computers or even entire computer networks perform different tasks, then they join together to create the program. Distributed computing allows problems to be solved that could not be solved on a single computer because of either the processing time or storage needs involved. Distributed computing allows much larger problems to be solved quicker than they could be solved using a single computer.

QUESTION: In what ways might distributed computing be better than sequential or parallel?

[Show answer/example](#)

Summary

[Sequential computing](#) is where the program runs one part after the other in order. [Parallel computing](#) is where two parts of the program run at the same time. Parallel computing doesn't necessarily go twice as fast as sequential computing, since code needs to be run to join the parts together. [Distributed computing](#) is where different networks work together to run a program.

Critical Thinking Questions

1. Imagine a huge online multiplayer video game. Why would it be impossible for this game to run effectively using only sequential computing on a single computer, and how do you think parallel or distributed computing would be essential for making it work?
2. Your school wants to speed up the process of grading thousands of standardized tests. If you could either assign one teacher to grade all tests individually (sequentially) or have multiple teachers grade different sections of the tests at the same time (in parallel), what are the advantages and disadvantages of each approach beyond just the total time saved?
3. Consider a task like building a complex Lego model. Some steps must be done one after another (like attaching a roof *after* the walls are built). Other steps could be done at the same time by different people (like one person building a car for the scene while another builds a tree). How does this Lego example illustrate why even in very fast "parallel" projects, there will always be some parts that must be done "sequentially"?

Questions (8)

1. True or False: Parallel computing is twice as fast as sequential computing

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

2. All of the code you will learn in this course is _____. Most beginners learn this mode of computing.

MULTIPLE CHOICE

Choose the correct answer:

- A. parallel
- B. sequential

3. True or False: Even parallel computing has sequential parts to the code

Choose the correct answer:

- A. True
- B. False

4. If a program took 6 minutes sequentially and 4 minutes in parallel, what would the speedup be?

MULTIPLE CHOICE

Choose the correct answer:

- A. $4/6$ or .66
- B. $6/4$ or 1.5
- C. 2
- D. -2

5. This kind of programming leverages many computers or even computer networks to run a program?

MULTIPLE CHOICE

Choose the correct answer:

- A. parallel
- B. sequential
- C. distributed
- D. uniform

6. What is the advantage of parallel computing

Choose the correct answer:

- A. It is easier to understand for beginners.
- B. It allows for the completion of tasks one at a time.
- C. It can make programs run faster.
- D. It has no limitations.

7. Why doesn't parallel computing always go twice as fast as sequential computing?

MULTIPLE CHOICE

Choose the correct answer:

- A. It requires more communication between processes.
- B. It uses a different programming language.
- C. It lacks beginning programmers.
- D. It has limited hardware support.

8. What is the "speedup" of a parallel solution

MULTIPLE CHOICE

Choose the correct answer:

- A. The time it takes to complete tasks in sequential computing divided by the time in parallel computing.
- B. The time it takes to complete tasks in parallel computing divided by the time in sequential computing.
- C. The time it takes to complete tasks simultaneously.
- D. The time it takes to write code.

Answer Keys & Solutions

Questions

1. True or False: Parallel computing is twice as fast as sequential computing

MULTIPLE CHOICE

Correct Answer:

A. True

✗ Incorrect

B. False

✓ Correct

Explanation:

Parallel computing takes extra time to sync different code parts together. It's still faster than sequential though.

2. All of the code you will learn in this course is _____. Most beginners learn this mode of computing.

MULTIPLE CHOICE

Correct Answer:

A. parallel

✗ Incorrect

B. sequential

✓ Correct

Explanation:

Sequential code is commonly what beginners learn.

3. True or False: Even parallel computing has sequential parts to the code

MULTIPLE CHOICE

Correct Answer:

A. True

✓ Correct

B. False

✗ Incorrect

Explanation:

Parallel computing takes extra time to sync different code parts together.

4. If a program took 6 minutes sequentially and 4 minutes in parallel, what would the speedup be?

MULTIPLE CHOICE

Correct Answer:

A. $4/6$ or .66

✗ Incorrect

B. $6/4$ or 1.5

✓ Correct

C. 2

✗ Incorrect

D. -2

✗ Incorrect

Explanation:

The speedup is sequential divided by parallel.

5. This kind of programming leverages many computers or even computer networks to run a program?

MULTIPLE CHOICE

Correct Answer:

A. parallel

✗ Incorrect

B. sequential

✗ Incorrect

C. distributed

✓ Correct

D. uniform

✗ Incorrect

Explanation:

This kind of computing is more spread than sequential or parallel.

6. What is the advantage of parallel computing

MULTIPLE CHOICE

Correct Answer:

A. It is easier to understand for beginners.

✗ Incorrect

B. It allows for the completion of tasks one at a time.

✗ Incorrect

C. It can make programs run faster.

✓ Correct

D. It has no limitations.

✗ Incorrect

Explanation:

Parallel programs run faster than sequential

7. Why doesn't parallel computing always go twice as fast as sequential computing?

MULTIPLE CHOICE

Correct Answer:

A. It requires more communication between processes.

✓ Correct

B. It uses a different programming language.

✗ Incorrect

C. It lacks beginning programmers.

✗ Incorrect

D. It has limited hardware support.

✗ Incorrect

Explanation:

Since several processes are happening at once, it needs instructions to join them together.

8. What is the "speedup" of a parallel solution

MULTIPLE CHOICE

Correct Answer:

A. The time it takes to complete tasks in sequential computing divided by the time in parallel computing.

✓ Correct

B. The time it takes to complete tasks in parallel computing divided by the time in sequential computing.

✗ Incorrect

C. The time it takes to complete tasks simultaneously.

✗ Incorrect

D. The time it takes to write code.

✗ Incorrect

Explanation:

The speedup measures how much faster it is to compute in parallel.

