Objective: Teach children about parallelization.

Age Level: Middle- to high-school age.

Background: Parallelization is when computers do multiple tasks at the same time instead of one after another. In everyday life, this can look like friends helping with party prep (one person vacuums, one shops, one makes the playlist) or several people painting different walls of a room at once.

In computing, the same principle is used to make programs run faster. Many modern devices have multiple processor "cores", which act like separate workers. Each core can handle part of a problem, so the task can be completed much faster. Cores make up a central processing unit (CPU), a component in every computer which is responsible for executing instructions. Parallelization used by pretty much every computing company there is, from Google to NASA to Roblox.

Before starting the activity, ask students to share times in their own lives when they've had to parallelize, such as in cooking, sports, or school projects. Then, explain that computers do the same thing, though sometimes splitting work adds *overhead* (delays). For example:

- Party planning: someone must drive to the shop before they can buy the food
- Cooking: ingredients must be peeled before they can be chopped
- Sports: moving between drills takes time

Materials Required:

- 16 small pieces of paper
- A writing utensil
- A larger piece of paper
- A timer
- Optional: Whiteboard & dry erase marker

Instructions:

1. Have a desk/table at the front of the classroom (or wherever you are doing the activity). If unavailable, just a general "work area" at the front is sufficient.
2. Write a one-digit number on each piece of paper. Make sure they're not all the same! Sum the numbers and make note of what this value is.
3. Start the first trial:
    a. Ask for a student volunteer to come up to the desk.
    b. Start the timer and have them sum all the numbers mentally (not aloud).
    c. When they're finished, have them write the sum on the larger piece of paper.

    d. Verify that this value is the sum you had before. If correct, stop the timer (if incorrect, restart the timer and have them re-do it or ask another student).
    e. Record how long this took & announce it to the class (optionally, write it on the whiteboard for the class to see). This is the time for one “core”.
4. Ask students how long they think it would take two cores. Will it take the same amount of time? Less? Half?
5. Now, do the second trial:
    a. Split the deck of papers into 2 roughly equal decks.
    b. Ask for 2 new student volunteers. Pick someone near the front desk, and someone a little farther away.
    c. Let them know that they will be summing the numbers in their pile, then writing it down on the paper.
    d. Start the timer, and give the student in the front the cards. Have them start summing their portion (mentally), writing their partial sum down on the larger piece of paper when done.
    e. Walk over and give the second student the other pile, saying something like. Have them also sum their numbers. However, when they’re done, ask them to walk over to the front desk and write their number down on the paper.
    f. Whichever student is done second will have to sum both their partial sum & the other student’s.
    g. Once you have a final sum, verify it, and stop the timer.
    h. Record & announce the time to the class; this is how long it took for two “cores”.
6. Talk about some observations:
    a. Was the second trial faster or slower?
    b. Did you expect it to take as long as it did? *Note: A popular guess here is that 2 cores will take half the time of 1 core. However, it likely took a little more than half.*
    c. What was the “overhead” here? *Answers to guide the students to, though they may come up with other valid potential overheads:*
        i. The students being far apart from one another
        ii. There only being one writing utensil
7. Ask students how long they think it would take four cores.
8. Finally, do the third trial:
    a. Split the deck into 4 piles. Make sure they are drastically uneven: perhaps one pile with just 1 paper, another with 3, another with 3, and another with 9. Don’t reveal to the class how unevenly split they are.
    b. Ask for 4 new student volunteers. Have them all come up to the front desk.
    c. Let them know that they will also be summing the numbers in their pile, then writing it down on the paper.
    d. Give each student a pile, face down, and tell them to not flip it until you start the timer.
    e. Start the timer.
    f. When all students are done & the sum has been verified, stop the timer.
    g. Record & announce the time to the class. This is the time for four cores.
9. Talk about some observations:

a. Was the time as expected?
b. What did the class observe when the students were adding?
c. What was the “overhead” here? *Answers to guide the students to, though they may come up with other valid potential overheads:*
    i. There only being one writing utensil
    ii. One person needs to sum all the other partial sums, adding to their workload
    iii. The pile of numbers being uneven
d. How could this system of 4 cores have been more optimal?

10. Discuss with students how they could make this system as optimal as possible, since we want our computers to be as fast as possible. *Some key points to touch on:*
    a. The “writing utensil” here is your device: what is going to display the outcome of the processes. Because of this, we can’t easily add more utensils.
    b. More cores doesn’t always mean better: imagine if we had 16 people, all with one number each: this would probably take even longer than just one person. So, how can we decide what is the most optimal number of cores?
    c. What if the cores could talk to each other? For example, in the 4 core trial, what if the people with the smaller piles summed their numbers together first, then wrote that, so that the final person only had to add 2 partial sums instead of 4?

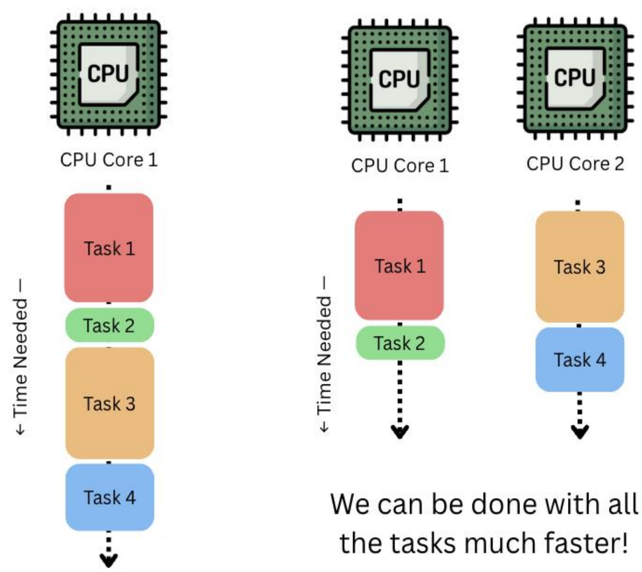
Notes: This lesson is inspired by one done at VT by Dr. Wu-Chun Feng. The graphic on the next page could also be used to introduce parallelization to students.

This is a CPU. It's the part in computers that's responsible for following directions, such as:

- "Play this YouTube video"
- "Calculate how much health I have"
- "Add a Tiktok filter to my face"

A CPU has multiple workers inside called "cores" that can work on different tasks at the same time. This teamwork approach is called **parallelization**.

Without parallelization: **With parallelization:**



We can be done with all the tasks much faster!