

Overview

This exercise aims to teach children a quick introduction to Amdahl's law, a law that helps define the overall speedup of an algorithm/process by parallelising parts of the process. The formula can be given as $1/(1-f)$, where f is the fraction of the program that has been sped up. In practice, Amdahl's law provides an estimate of the overall speed at which the algorithm can be executed.

Suitable For

This exercise is the version that is suitable for students studying Higher Computing, or Advanced Higher students.

Key Concepts

Amdahl's law, parallelisation, speedup, multi-core architecture

Learning Outcomes

- Understand how parallelising parts of a program can help speed up the execution.
- Understand that allowing multiple cores to spread out the processing of a program decreases execution time
- Understand the importance of parallelism
- Understand how Amdahl's law estimates the overall speedup of a program after parallelisation.

Success Criteria

- I can understand how parallelisation is an important factor in speeding up a program
- I can derive Amdahl's law (Potentially?)
- I can explain the need for multiple cores when computing.

Time Required

1 period - 1 hour

Preparation

Print out the items given in the pack, one of each.

Prior Learning Assumed

Perhaps the idea of multiple cores and parallelism might already be known to them - this could solidify the idea.

Outline of Activity

1. Explain to the class that we are going to be looking at a new topic for computing, called parallelism.
2. Show the board and explain that this is what a model computer might look like inside, with its multiple cores - which are essentially like little computer brains. Explain that this is an example of a four core computer.
3. Outline that say for example you want to want to dig a very big hole. One man might take 60 minutes to dig this big hole, and that this is like giving one core all the work to do. It's going to be really really slow and difficult to do.

4. Ask the class for any feedback on what they might do instead? Some options might be:
 - a. Get more men involved to share out the task
 - b. Dig a smaller hole - maybe, but we need this hole to be this size!
5. Suggest that the concept of getting more men involved to share out the task is correct - this idea is called parallelism!
6. Explain that essentially you can turn a program from a serial one (all in one core) to a program that has a parallelisable section and a non-parallelisable section.
7. Introduce the program (long strip of pink paper) and it should be laid out along the top core, in one line. This is a serial program as it is all running in one core.
8. Cut up the program into two sections and show that one section is the parallelisable section of the program. This section can be shared out among the other processors.
9. Place the parallelisable section under the second core, and explain that this has shown that it has saved us time by pointing out along the bottom that the time has been reduced.
10. Finish by saying that this is the main idea of parallelism.

POTENTIAL EXTRA

Outline that there is a function that can help figure out how quick a program could be if it was parallelised. This is called Amdahl's law.

You can move to the board to demonstrate the following points:

- The total time of the whole program to run is T .
- The total time of the non-parallelisable part is B .
- Thus, the total time of the parallelisable part is $T-B$.
- If there are N cores running the parallelisable part, this means that the fastest the parallelisable part can be run at is $(T-B)/N$.
- This means, the fastest total time of the whole program running on N cores is:
 - $T(N) = B + (T-B)/N$
 - (Adding in the non-parallelisable section)
- This is Amdahl's law.

You could draw out this diagram to further explain.

