Parallelism

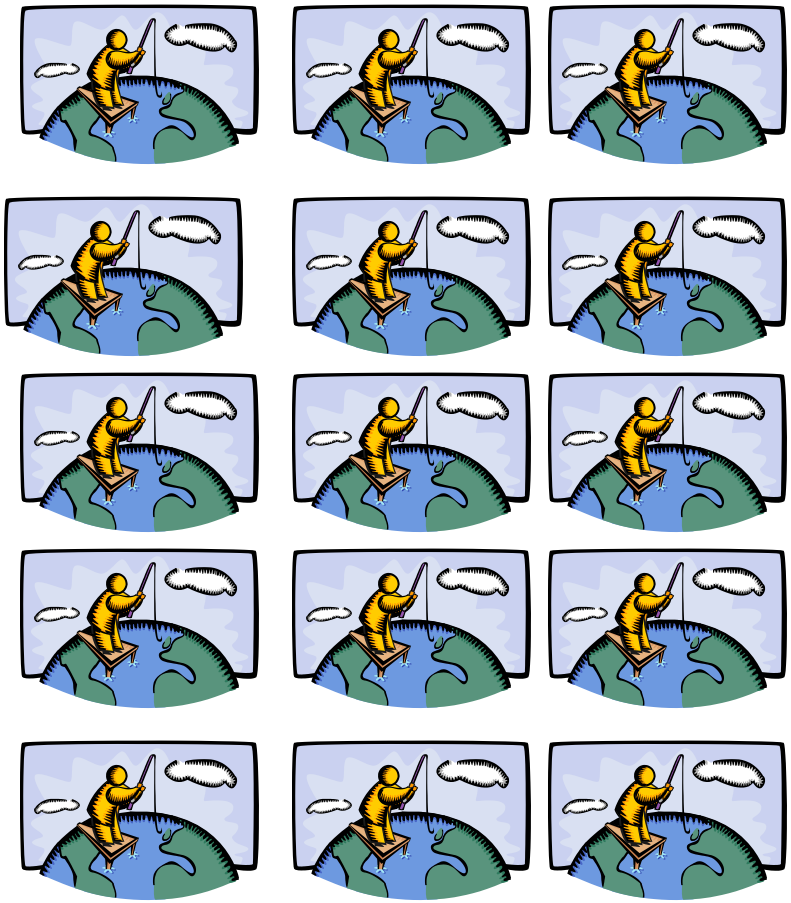
Robert Whitten Jr

Major props to Henry Neeman at Oklahoma University and his Supercomputing in Plain English

Parallelism

Parallelism means doing multiple things at the same time: you can get more work done in the same time.

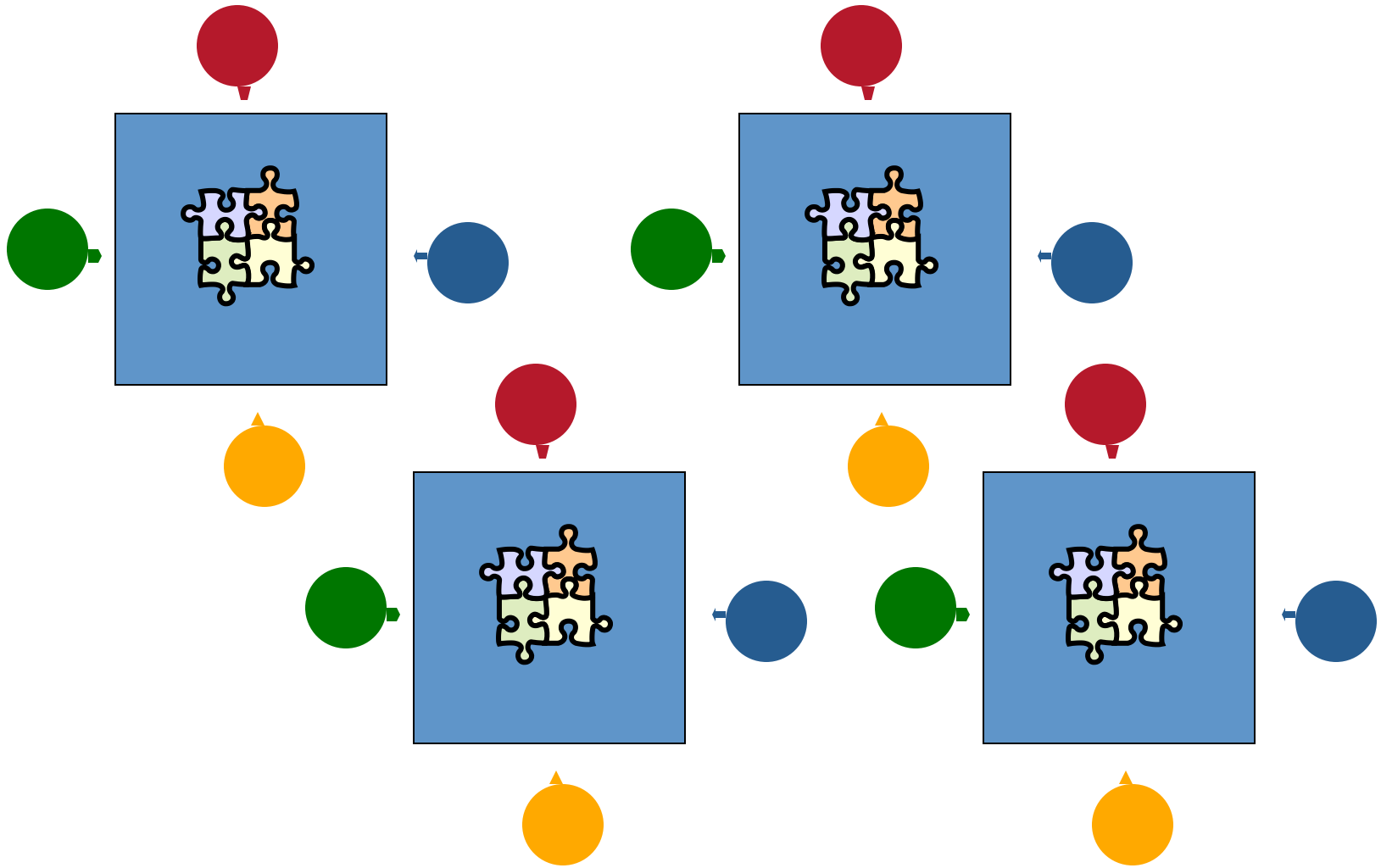
Less fish …



More fish!

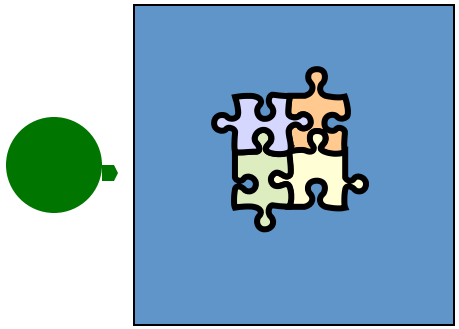
Oak Ridge National Laboratory

The Jigsaw Puzzle Analogy



Oak Ridge National Laboratory

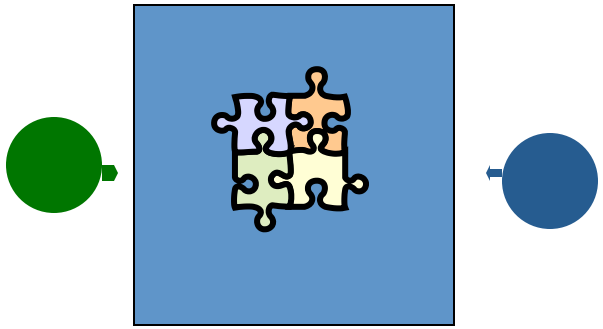
Serial Computing



Suppose you want to do a jigsaw puzzle that has, say, a thousand pieces.

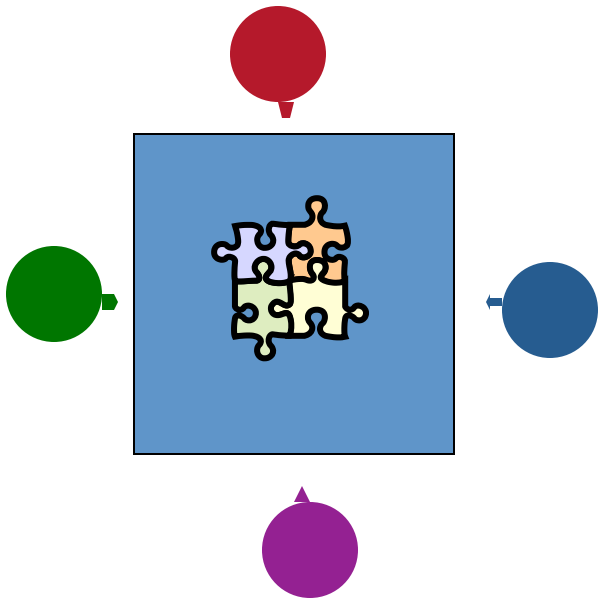
We can imagine that it'll take you a certain amount of time. Let's say that you can put the puzzle together in an hour.

Shared Memory Parallelism



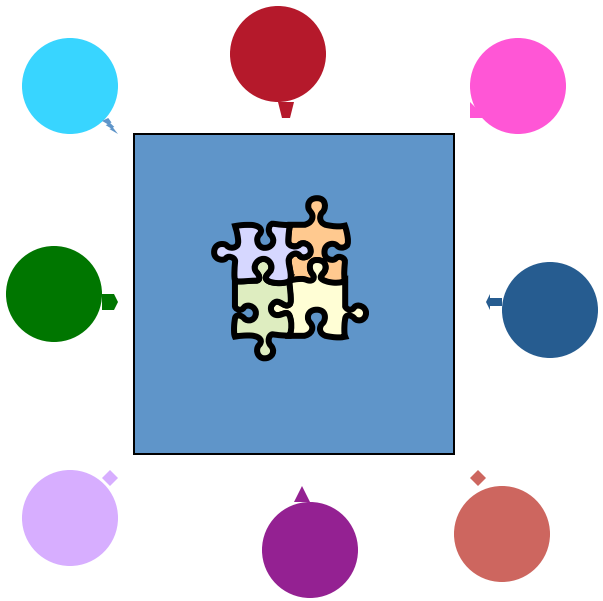
If someone sits across the table from you, then he can work on his half of the puzzle and you can work on yours. Once in a while, you'll both reach into the pile of pieces at the same time (you'll ***contend*** for the same resource), which will cause a little bit of slowdown. And from time to time you'll have to work together (***communicate***) at the interface between his half and yours. The speedup will be nearly 2-to-1: you all might take 35 minutes instead of 30.

The More the Merrier?



Now let's put two more people on the other two sides of the table. Each of you can work on a part of the puzzle, but there'll be a lot more contention for the shared resource (the pile of puzzle pieces) and a lot more communication at the interfaces. So you will get noticeably less than a 4-to-1 speedup, but you'll still have an improvement, maybe something like 3-to-1: the four of you can get it done in 20 minutes instead of an hour.

Diminishing Returns



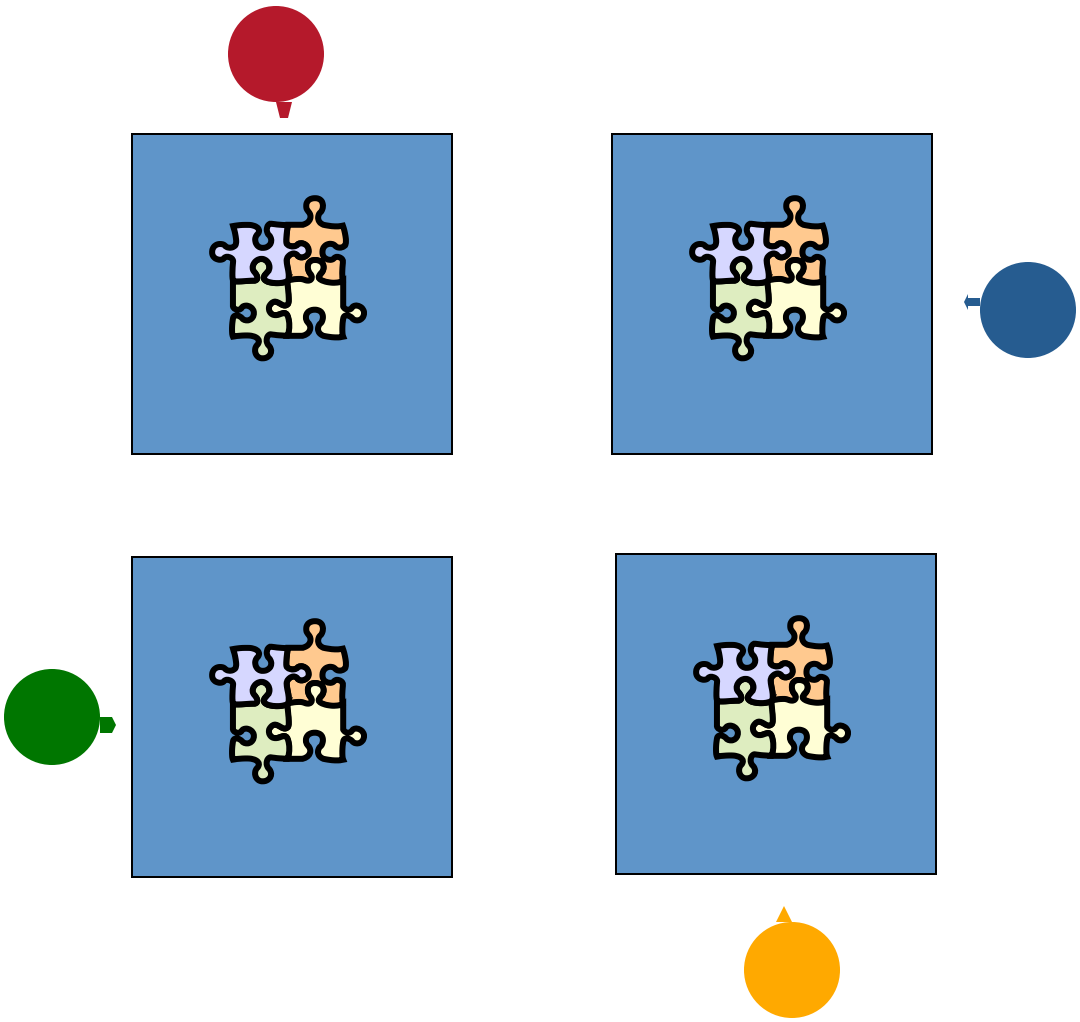
If we now put four more people on the corners of the table, there's going to be a whole lot of contention for the shared resource, and a lot of communication at the many interfaces. So the speedup you will get will be much less than we'd like; you'll be lucky to get 5-to-1.

So we can see that adding more and more workers onto a shared resource is eventually going to have a diminishing return.

Distributed Parallelism

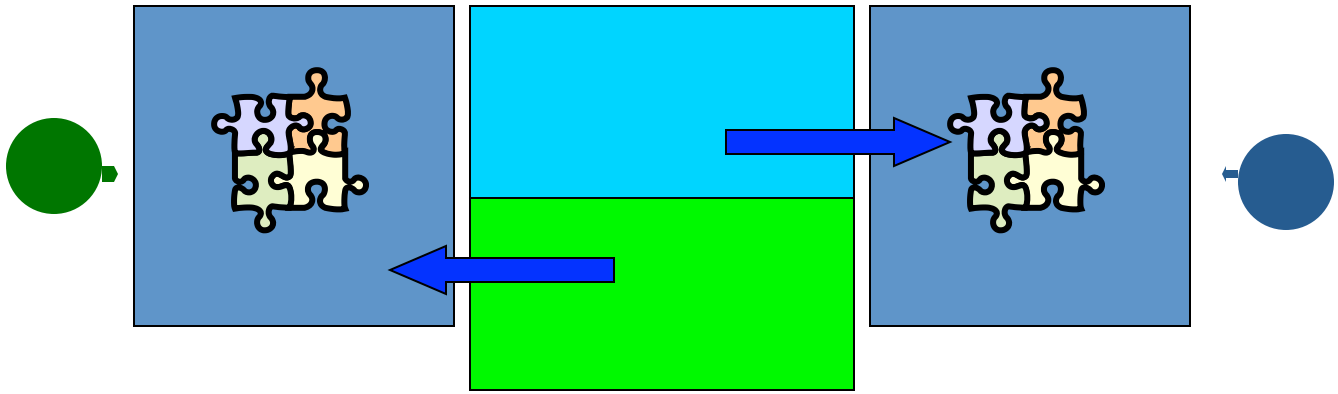
Now let's try something a little different. Let's set up two tables, and let's put you at one of them and someone else at the other. Let's put half of the puzzle pieces on your table and the other half of the pieces on the other. Now you all can work completely independently, without any contention for a shared resource. **BUT**, the cost of communicating is **MUCH** higher, and you need the ability to split up (***decompose***) the puzzle pieces reasonably evenly, which may be tricky to do for some puzzles.

More Distributed Processors



It's a lot easier to add more processors in distributed parallelism. But, you always have to be aware of the need to decompose the problem and to communicate between the processors. Also, as you add more processors, it may be harder to ***load balance*** the amount of work that each processor gets.

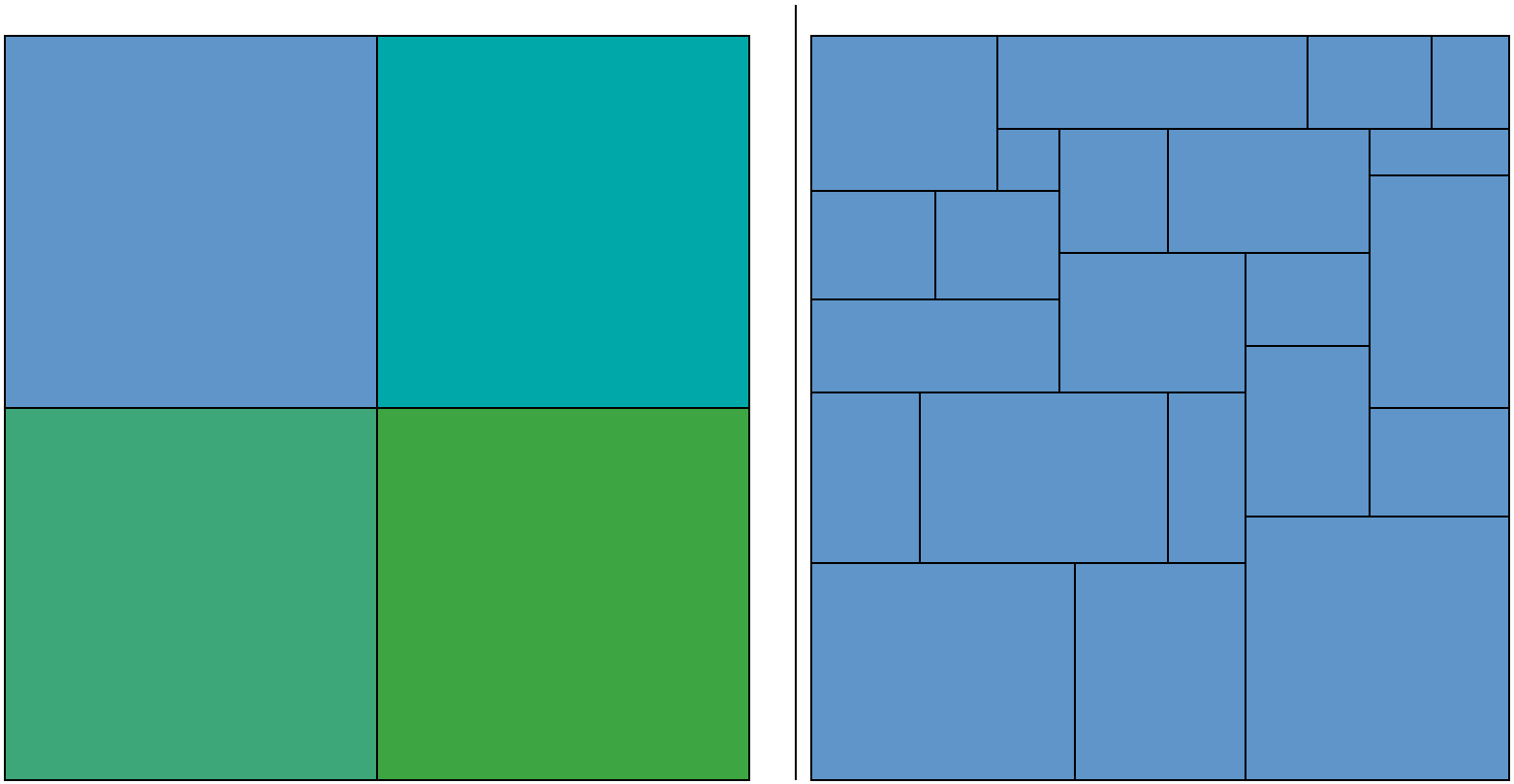
Load Balancing



<u>Load balancing</u> means giving everyone roughly the same amount of work to do.

For example, if the jigsaw puzzle is half grass and half sky, then you can do the grass and someone can do the sky, and then you only have to communicate at the horizon – and the amount of work that each of you does on your own is roughly equal. So you'll get pretty good speedup.

Load Balancing



Load balancing can be easy, if the problem splits up into chunks of roughly equal size, with one chunk per processor. Or load balancing can be very hard.

Concepts of Parallelism

- Shared memory architecture
- Distributed memory architecture
- Contention
- Communication
- Load balancing
- Decomposition

Oak Ridge National Laboratory

pThreads

- Parallelization on a single CPU
- pthread = posix thread

Oak Ridge National Laboratory

Questions?

http://www.olcf.ornl.gov