Data Size and Instruction Count Matters

Let's learn about factors which improve the performance of parallel algorithms.

WE'LL COVER THE FOLLOWING

Optimizing the Use of Parallel Algorithms

How to get the best performance with parallel algorithms?

You need two things:

- a lot of data to process
- instructions to keep CPU busy

What's more, we have to remember one rule:

> In general, parallel algorithms do more work, as they introduce an extra cost of managing the parallel execution framework as well as splitting tasks into smaller batches.

Parallel vs. Serial

First of all, we have to think about the size of the data we operate on. If we have only a few files, with a few dozen records, then it might happen that parallel execution won't be faster than the serial version. But if we have lots of files, with hundreds of lines each, then the potential might increase.

The second thing is the instruction count. CPU cores need to compute and not just wait on memory. If your algorithms are memory-bound, then parallel execution might not give any speed-up over the sequential version. In our

case, it seems that the parsing strings task is a good match here. The code

performs searching on strings and does the numerical conversions, which keeps CPU busy.

In the next lesson, we will look at parallel data conversion.