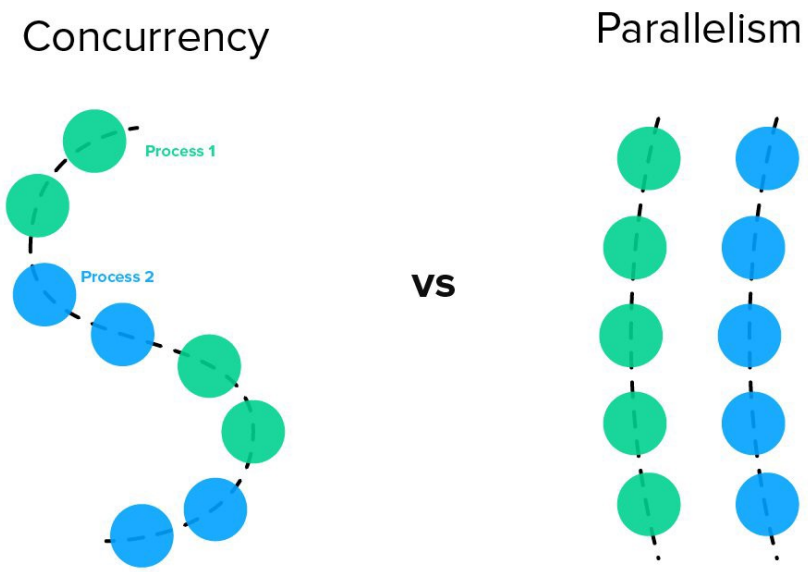
kwahome.medium.com

Concurrency is not Parallelism - Wahome - Medium

Wahome

3 minutes



concurrency vs parallelism

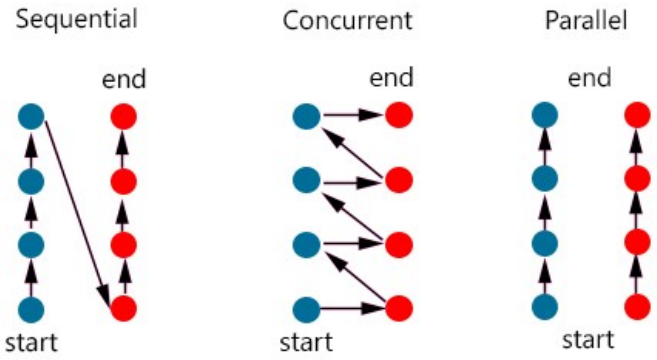
> “Concurrency is about dealing with lots of things at once. Parallelism is about doing lots of things at once.” — Rob Pike

In programming, concurrency is the composition of independently executing processes, while parallelism is the simultaneous execution of (possibly related) computations.

A concurrent program has multiple logical threads of control. These threads may or may not run in parallel.

A parallel program potentially runs more quickly than a sequential program by executing different parts of the computation simultaneously; in parallel. It may or may not have more than one logical thread of control.

Concurrency enables Parallelism.



sequential vs concurrent vs parallel

“Concurrency is about structure, parallelism is about execution.”

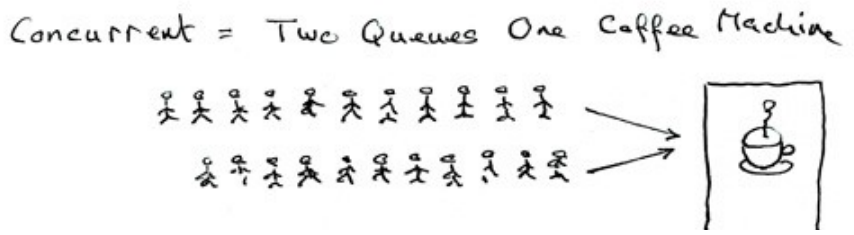
Concurrency provides a way to structure a solution to solve a problem that may (but not necessarily) be parallelizable.

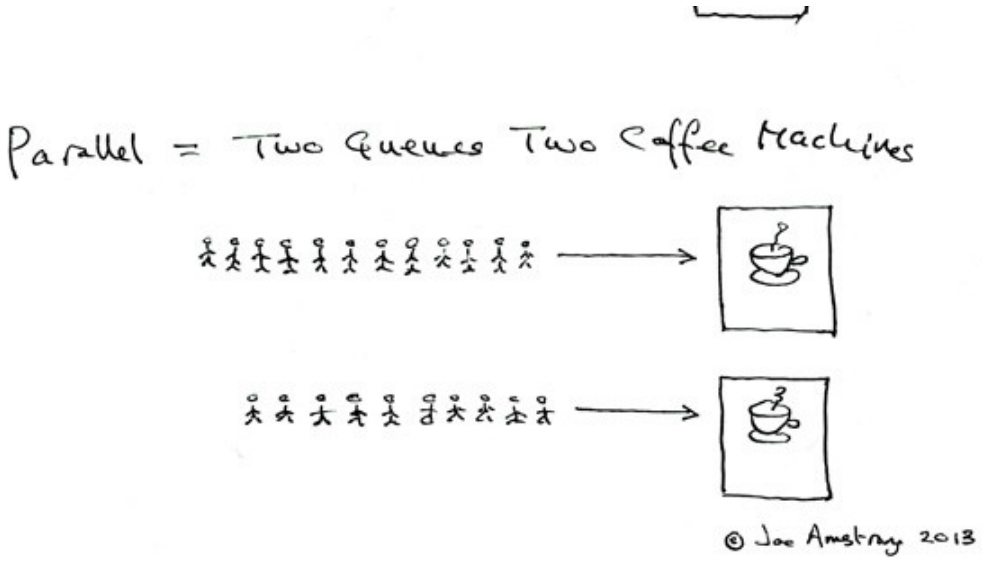
The modern world is parallel. It has:

- Multicores
- Networks
- Clouds of CPUs
- Loads of users

Concurrency makes parallelism easy.

An analogy

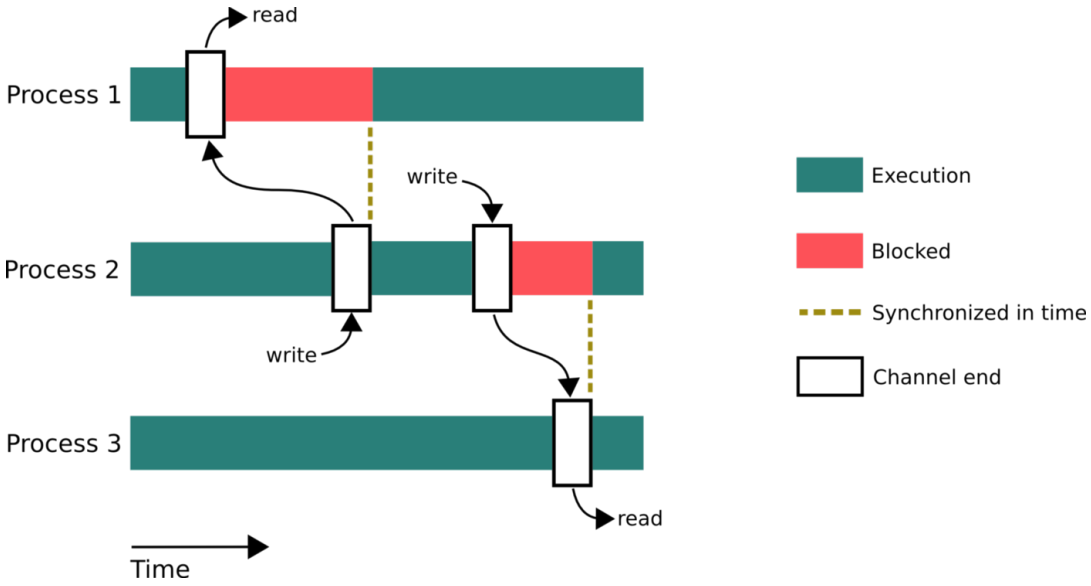




Concurrent: Two queues to one coffee machine

Parallel: Two queues to two coffee machines

Communicating Sequential Processes



CSP illustration

Communicating Sequential Processes (CSP) is a mathematical notation for describing patterns of interaction.

C. A. R. Hoare in his 1978 paper, suggests that input and output are basic primitives of programming and that parallel composition of communicating sequential processes is a fundamental program structuring method. When combined with a development of

Dijkstra's guarded command, these concepts become surprisingly versatile.

Communication is the means to coordinate the independent executions and should be favoured as a collaboration mechanism over shared state. CSP is the model on which Go concurrency (and others like Erlang) is based on.

Summary

Concurrency is about structure, parallelism is about execution.

Concurrency enables parallelism.

Communication is the means to coordinate independent executions and should be favoured as a collaboration mechanism over shared state.

Thank you for reading. I sincerely hope it was a nice read.

You can catch me at:

GitHub: kwahome

Twitter: @kwahome_

References

1. https://talks.golang.org/2012/waza.slide#10
2. https://www.cs.cmu.edu/~crary/819-f09/Hoare78.pdf
3. https://wiki.tcl-lang.org/page/Dijkstra%27s+guarded+commands
4. https://vimeo.com/49718712