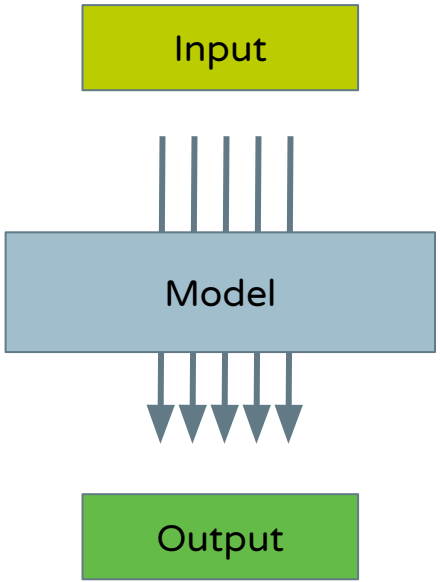
Parallel Computing

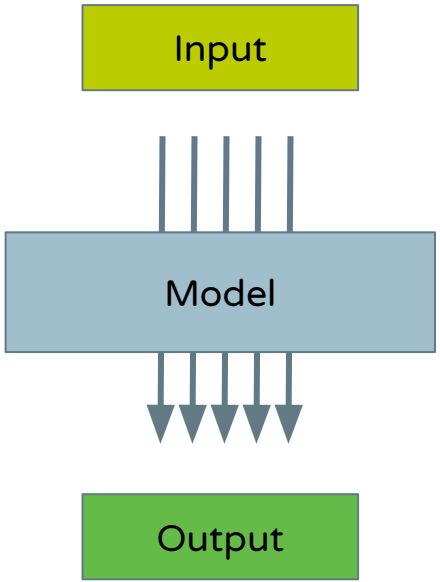
What is it and when to use it

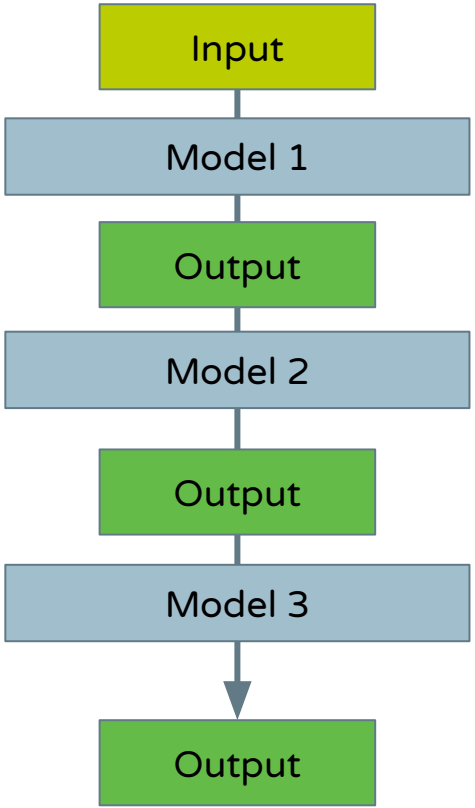
Types of Computations

Outputs are independent of each other



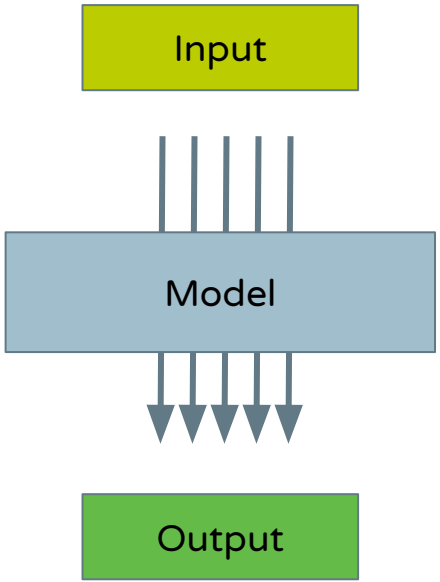
Types of Computations

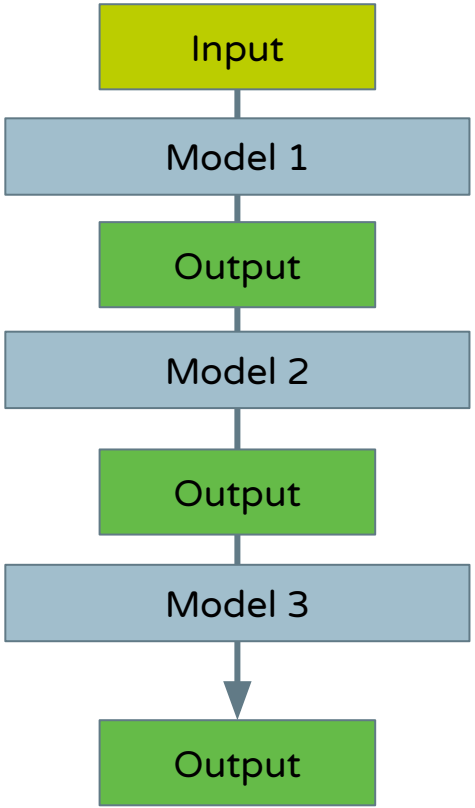




Types of Computations

Parallel Problem: Outputs are independent of each other

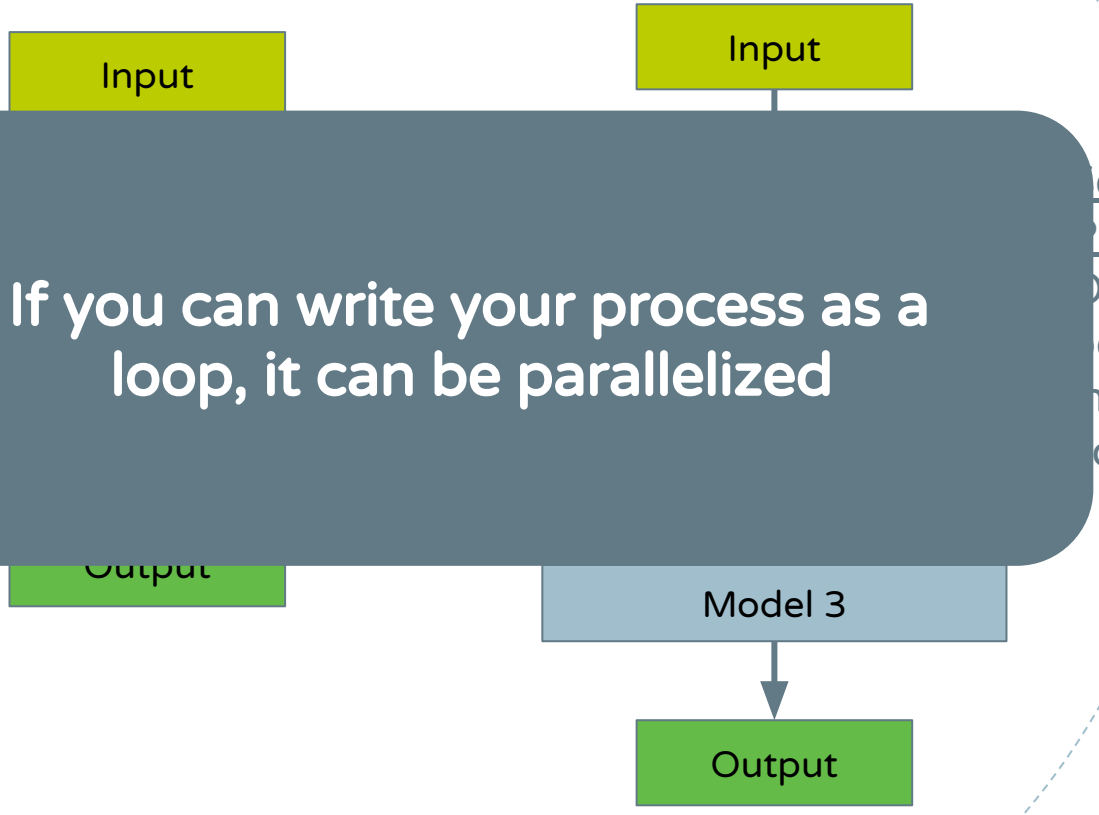




Serial Problem: Output becomes the input for next computation

Types of Computations

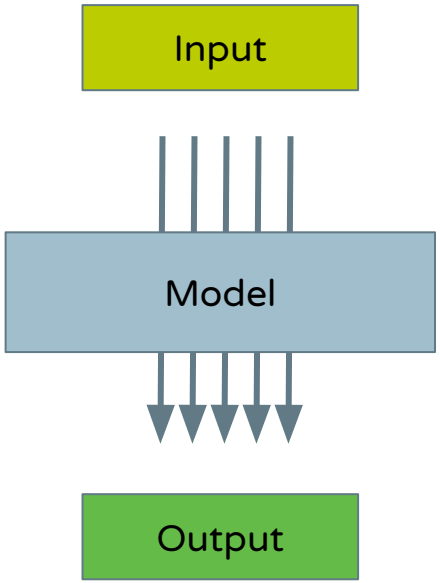
Parallel Problem: Outputs are independent of each other

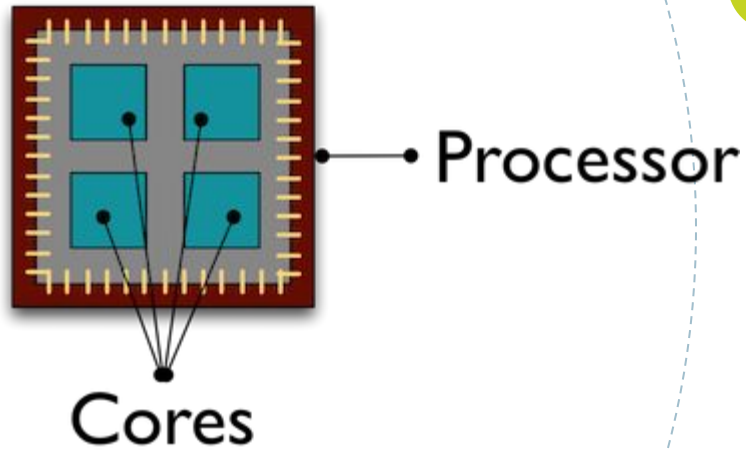


If you can write your process as a loop, it can be parallelized

erial Problem: Output becomes the nput for next omputation

Each core can run a separate calculation





Standard computers often have 4-8 cores

Sockets

Launches R on new core and manually exports functions, objects, packages, etc.

- Works on all OS
- More flexible/complicated to implement

Forking

Creates a copy of the current R instance on new core

- Does not work on Windows
- Faster
- Can be unstable/odd behavior

Two Common Packages

parallel

-feels like an apply statement

- some functionality built into base R

-mclappy(), but is treated as lapply on Windows

doParallel

-feels like loops

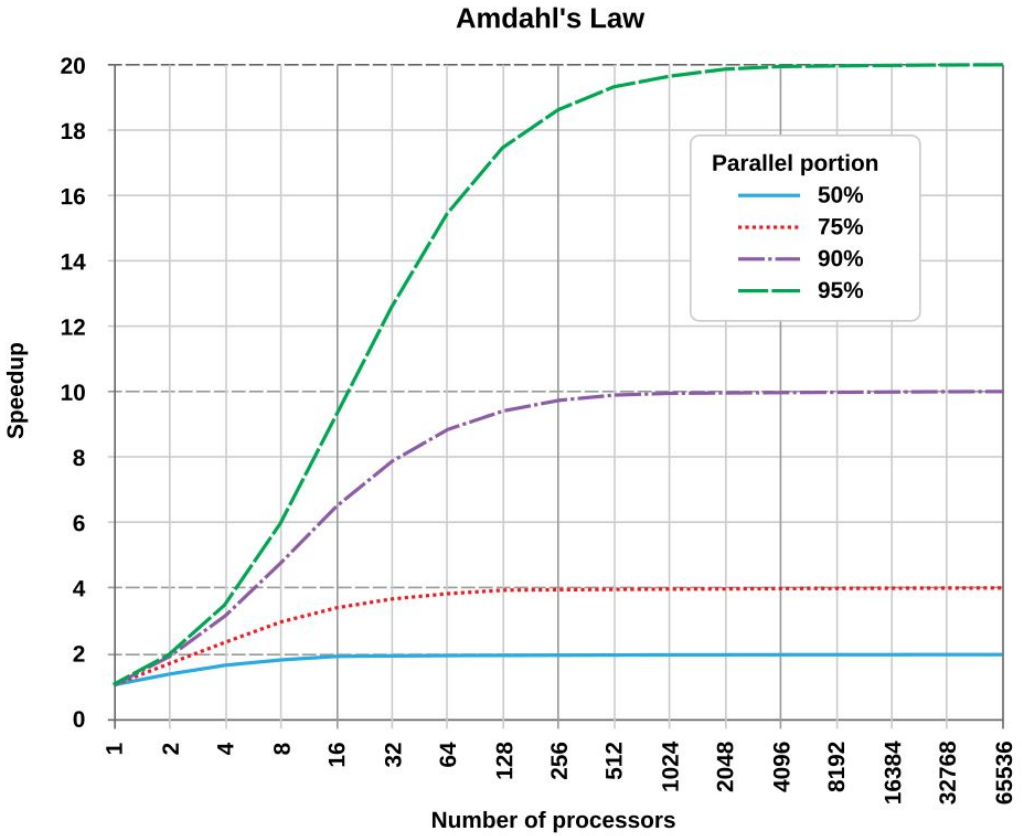
-uses the foreach package

- %dopar%

* Choosing approach is dependent on operating system and coding preference

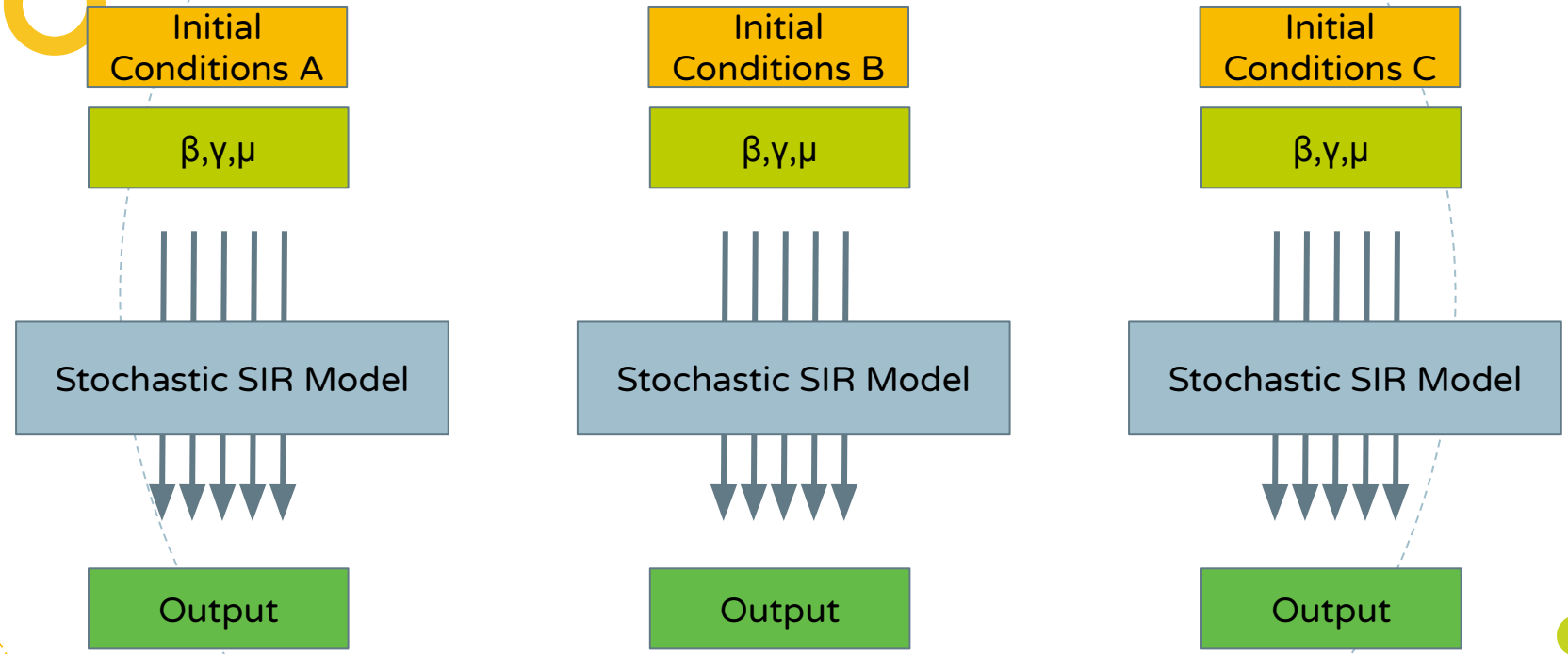
2x Cores ≠ 2x Speed

There is a computational cost to manage additional cores



Amdahl's law - Wikipedia

My common use case



Further readings

Vignettes:

Getting Started with doParallel and foreach

Package `parallel'

Applied example of parallel package: Stats 506: Parallel Processing

Comparison of parallel and doParallel: 23 Additional Resources: Parallel Computing in R | Reproducible Research Techniques for Synthesis