Lecture 15 - Introduction to Parallelism

DSE 512

Drew Schmidt
2022-03-22

From Last Time

- Homework 2 is graded
- Homework 3 posted tonight some time
- Let's talk about that...

Where We've Been

Module 1: Basic Cloud and HPC

- Lecture 1 - Introduction
- Lecture 2 - Overview of HPC and the Cloud
- Lecture 3 - Introduction to Remote Computing
- Lecture 4 - Introduction to Containers
- Lecture 5 - Introduction to ISAAC
- Lecture 6 - MPI and Singularity

Where We've Been

Module 2: Performance Optimization

- Lecture 7 - Introduction to Performance Optimization
- Lecture 8 - High Level Language Optimizations
- Lecture 9 - Computational Linear Algebra Part 1
- Lecture 10 - Computational Linear Algebra Part 1
- Lecture 11 - GPGPU (The Easy Parts) Part 1
- Lecture 12 - GPGPU (The Easy Parts) Part 2
- Lecture 13 - Utilizing Compiled Code
- Lecture 14 - I/O

Where We're Headed

Module 3: Parallelism

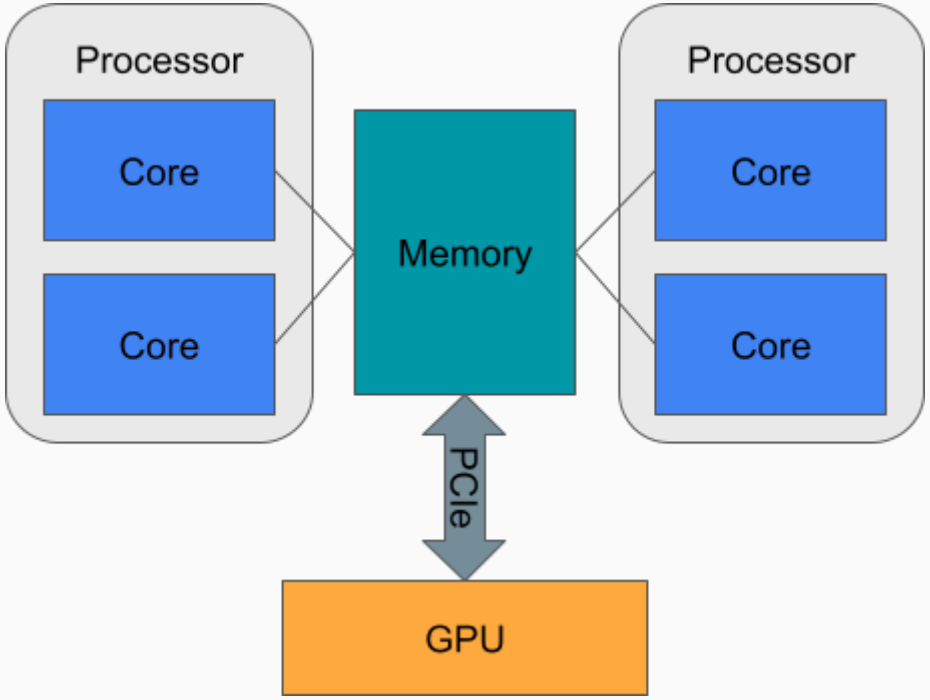
- Introduction to Parallelism
- Forks and Threads
- MPI
- MapReduce

Intro to Parallelism

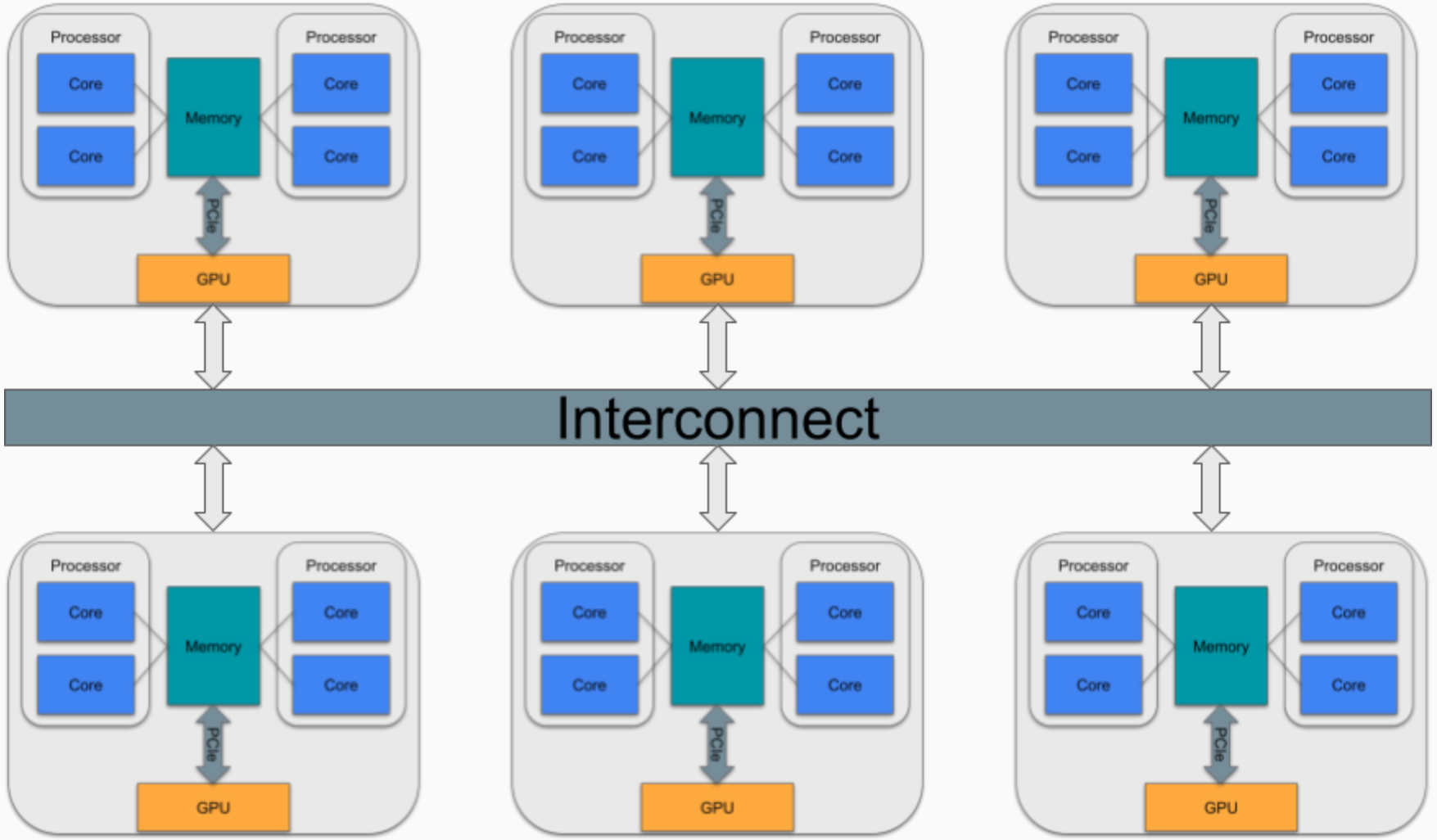
Parallelism

- We've seen some before!
- Treated mostly as a "black box"
- We'll get more familiar with implementing parallel algorithms

Hardware

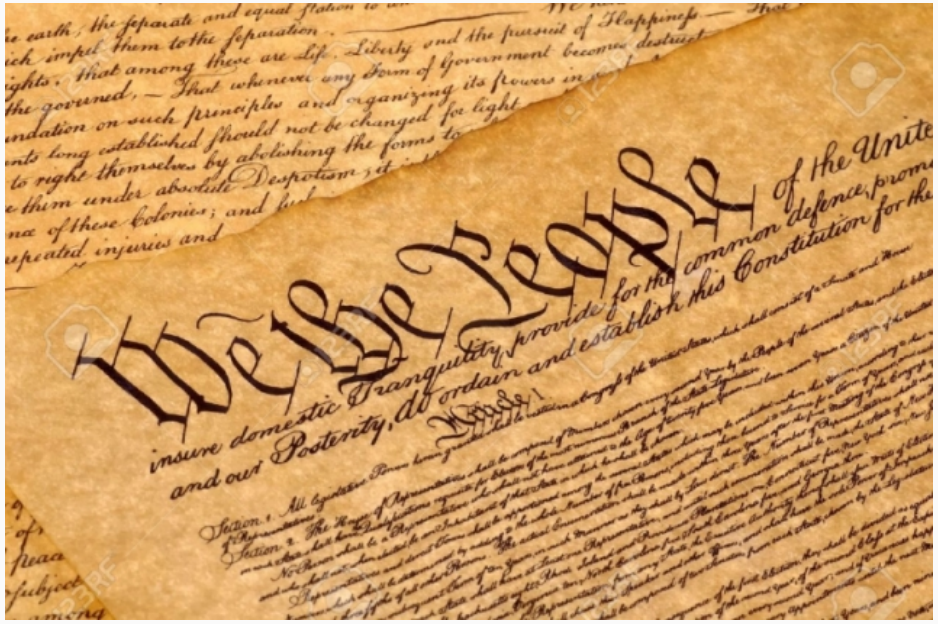


Hardware

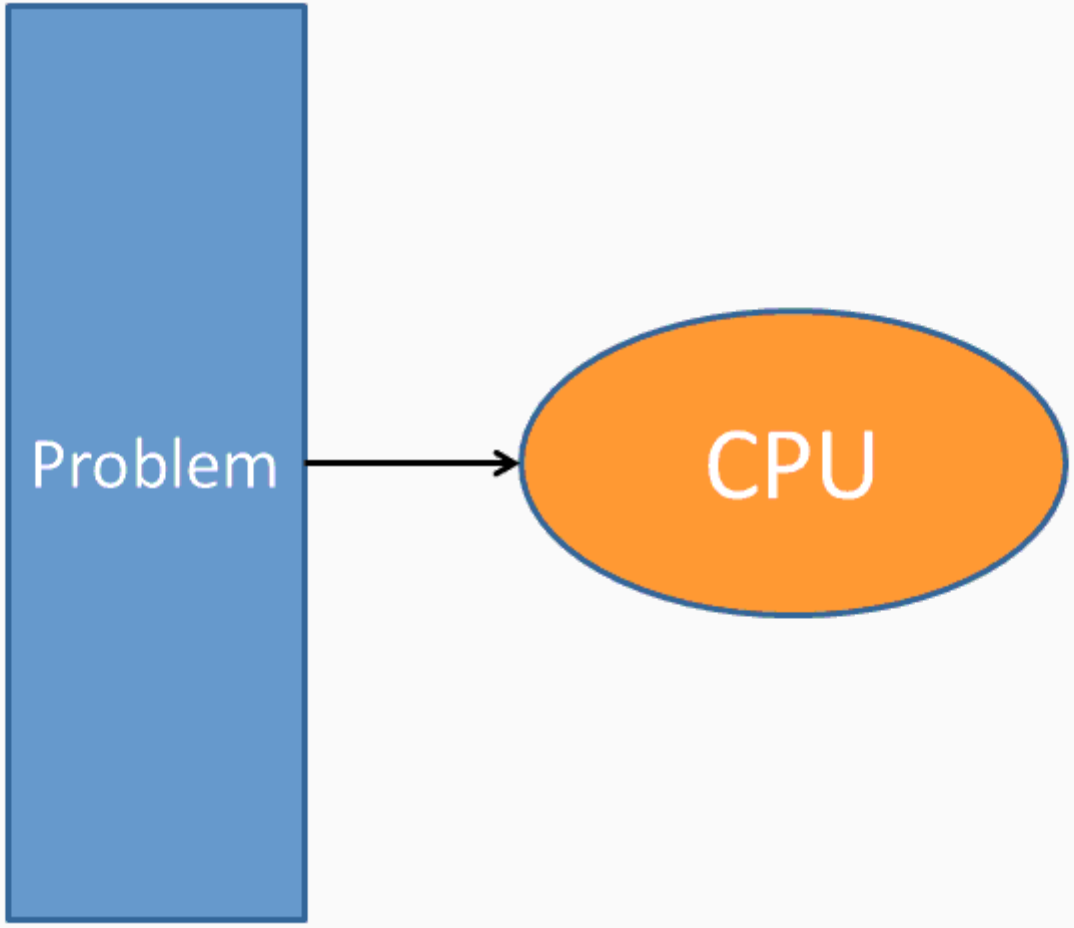

Processor
Core
Core
Memory
Processor
Core
Core
PCIe
GPU
Interconnect

What Is Parallelism?

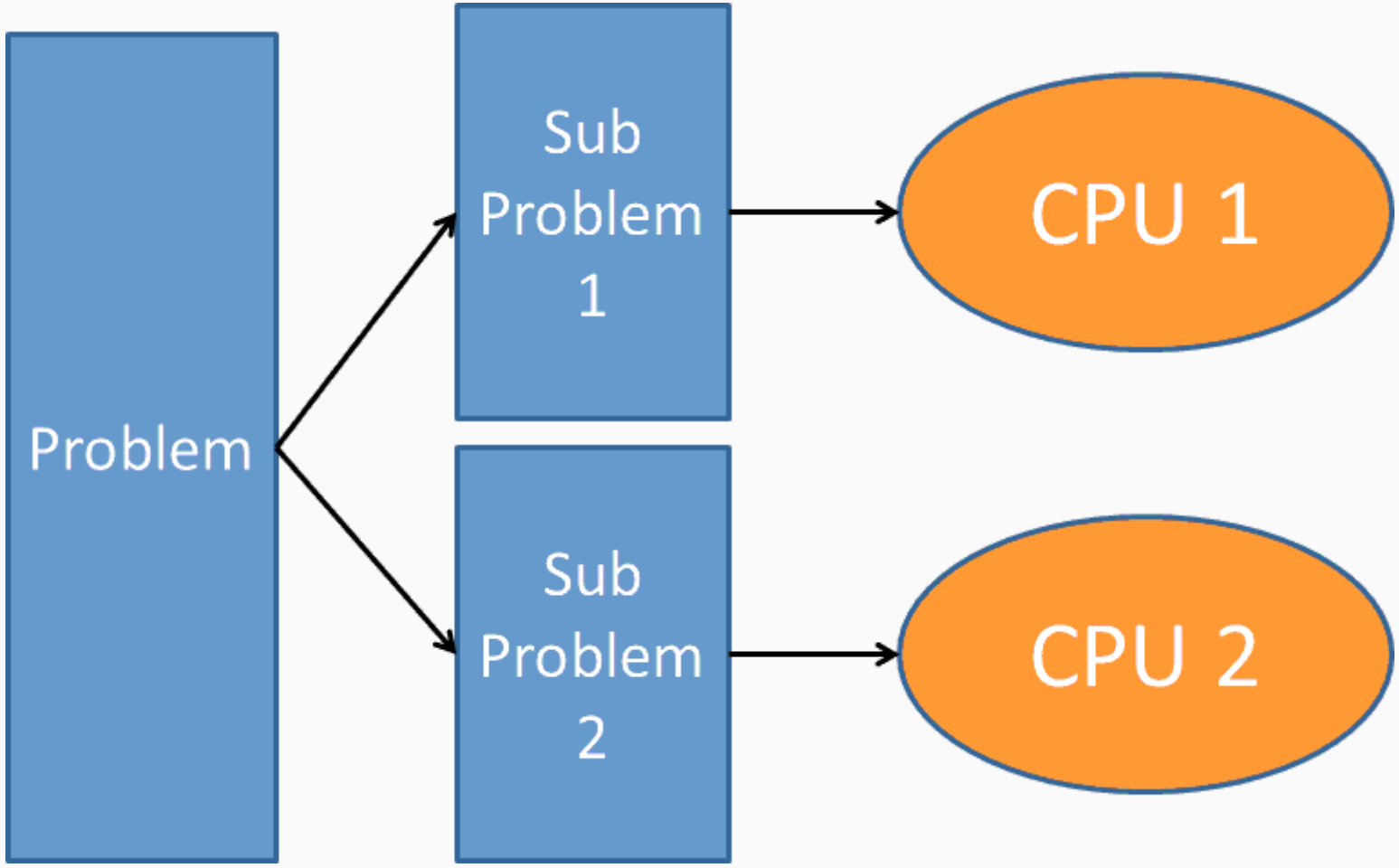
Parallelism is about *independence*.



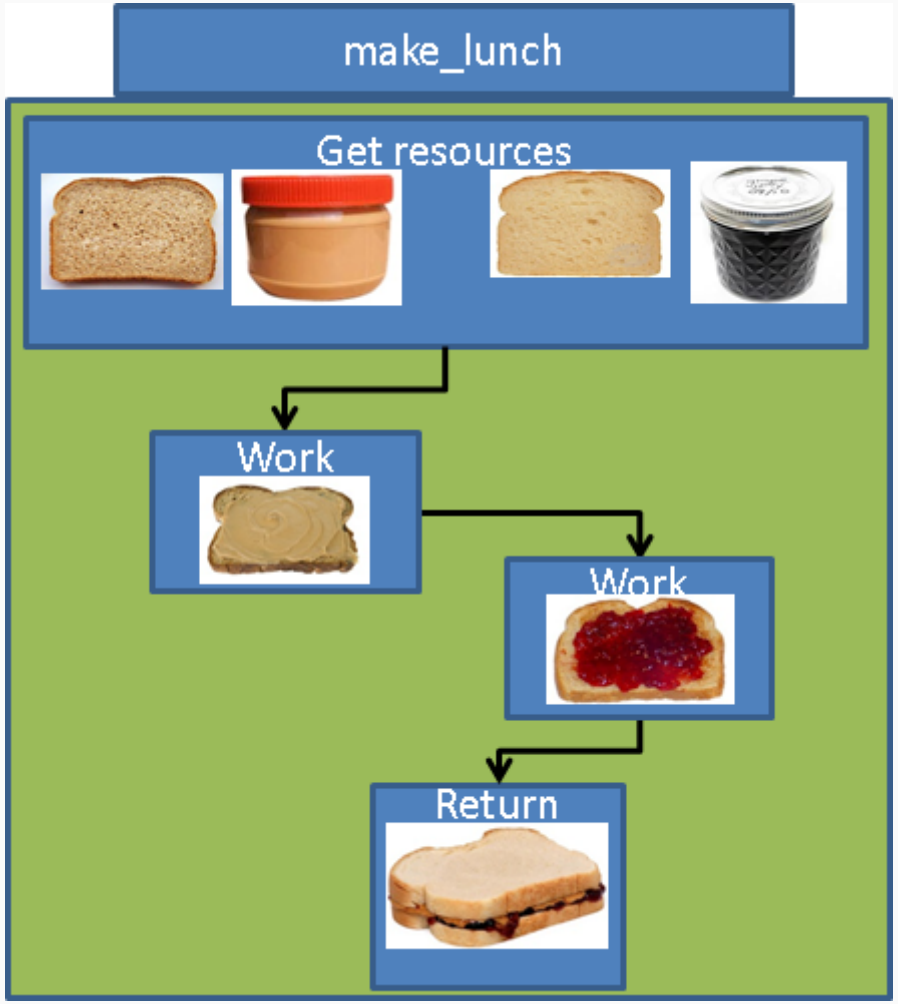
Serial Programming



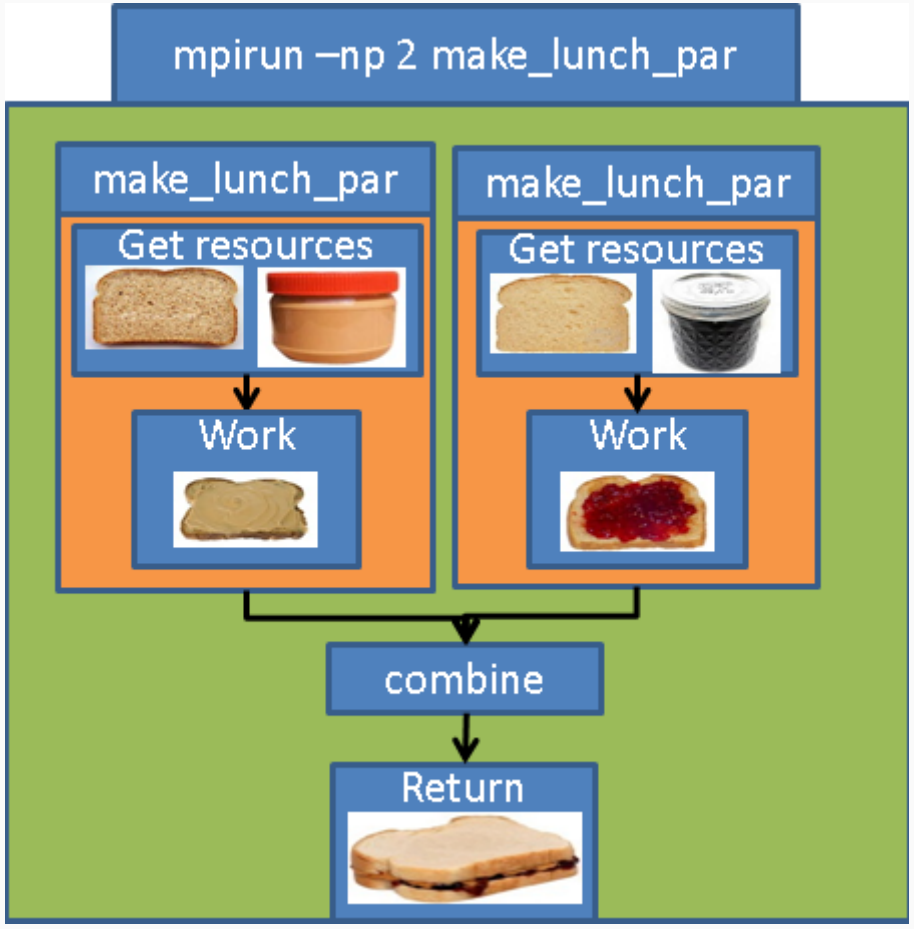
Parallel Programming



Make Lunch: Serial



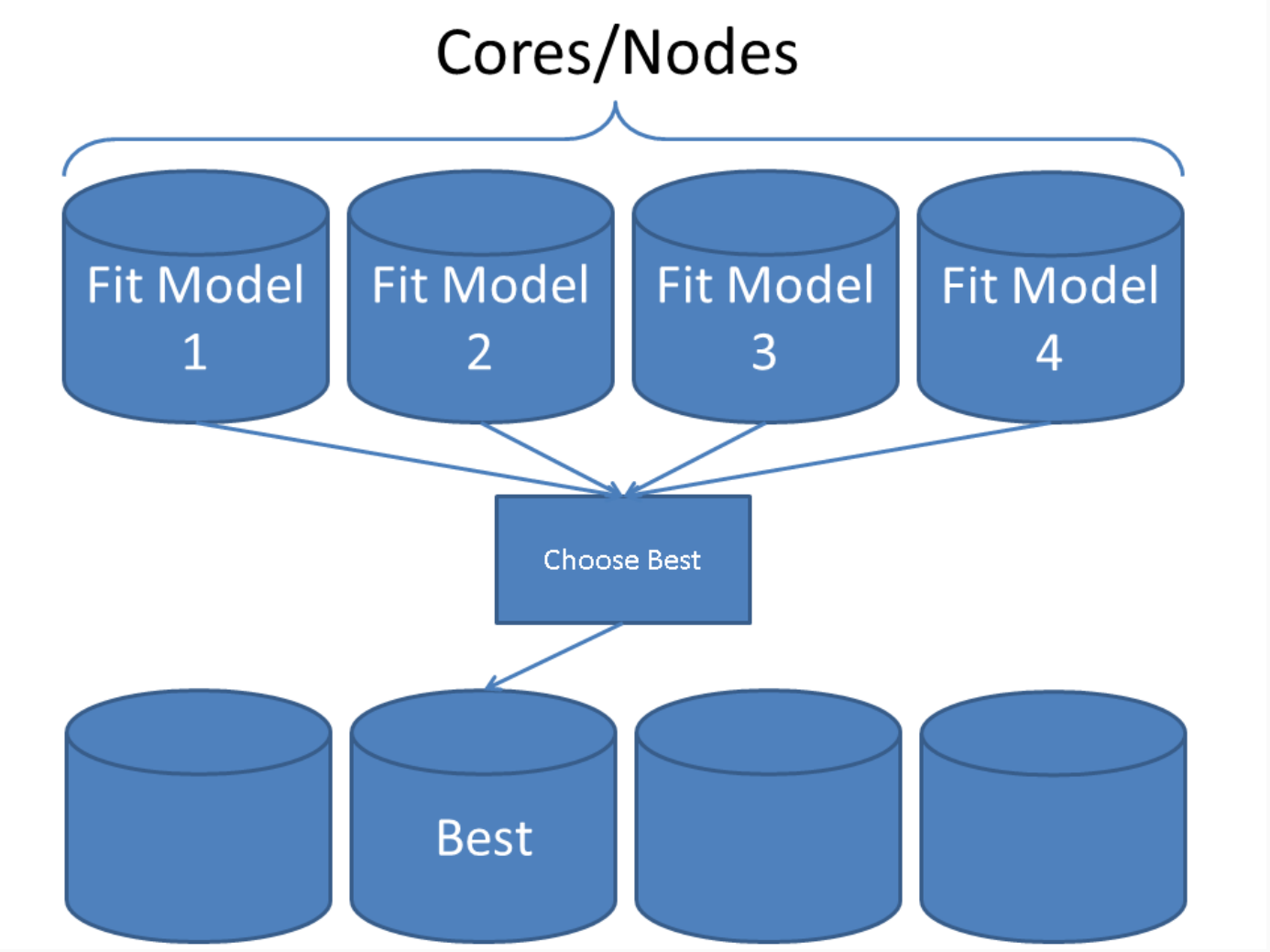
Make Lunch: Parallel



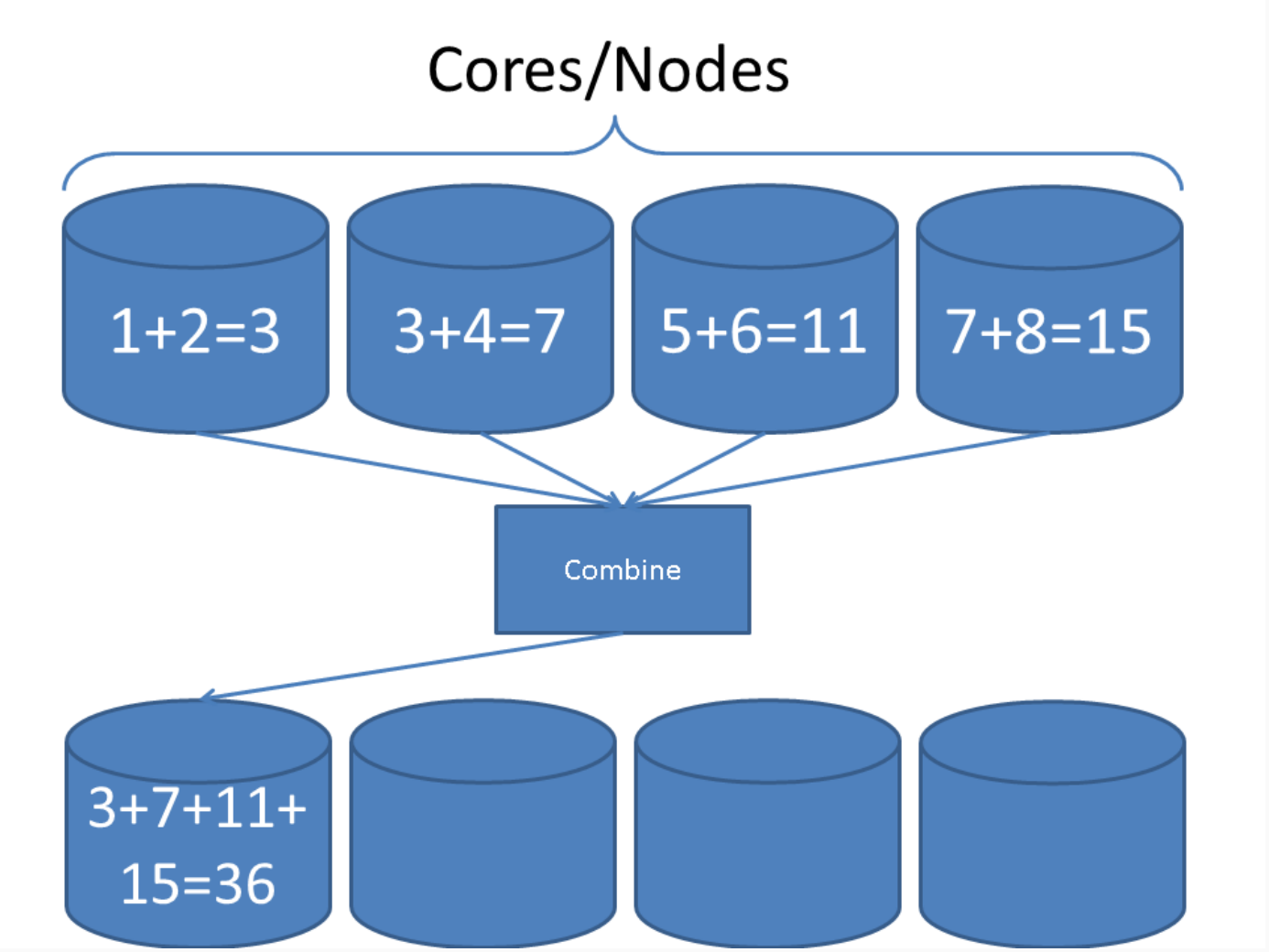
Kinds of Parallelism

- Task parallelism
  - *Very important to data analysis pipelines*
  - Example: fitting multiple models, choosing the best one
- Data parallelism
  - Less common pattern in data science
  - Example: "Big data"

Task Parallelism

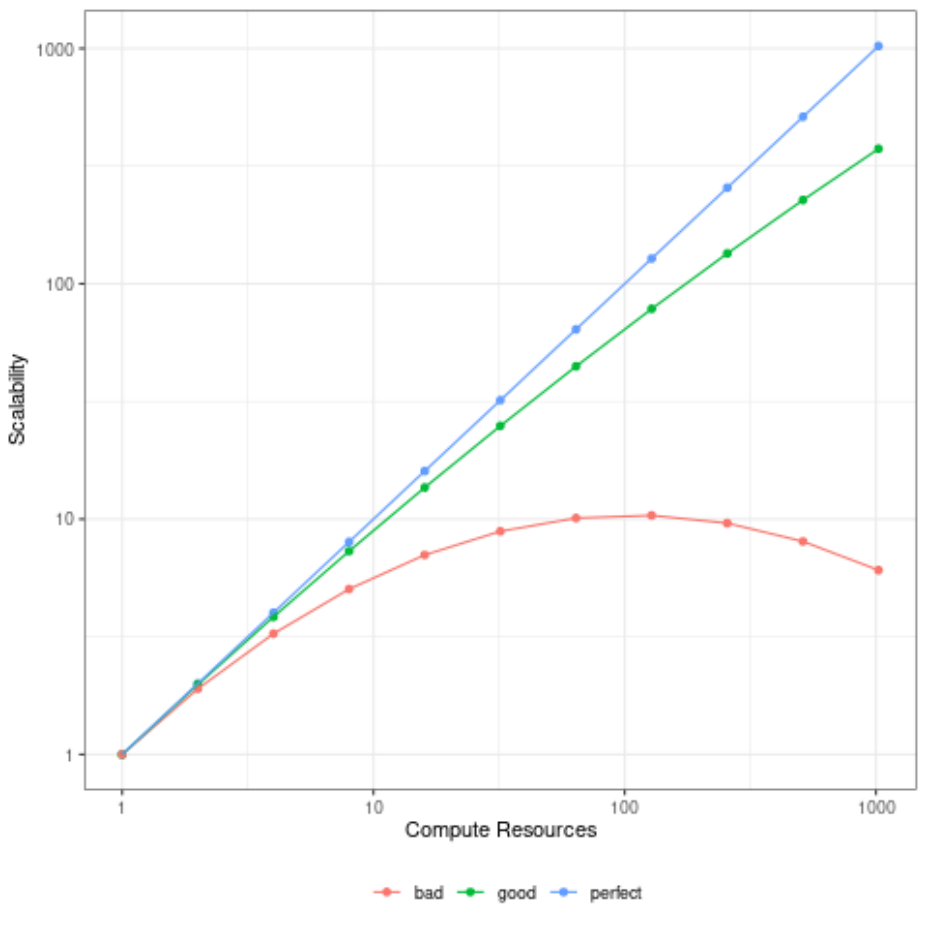


Data Parallelism

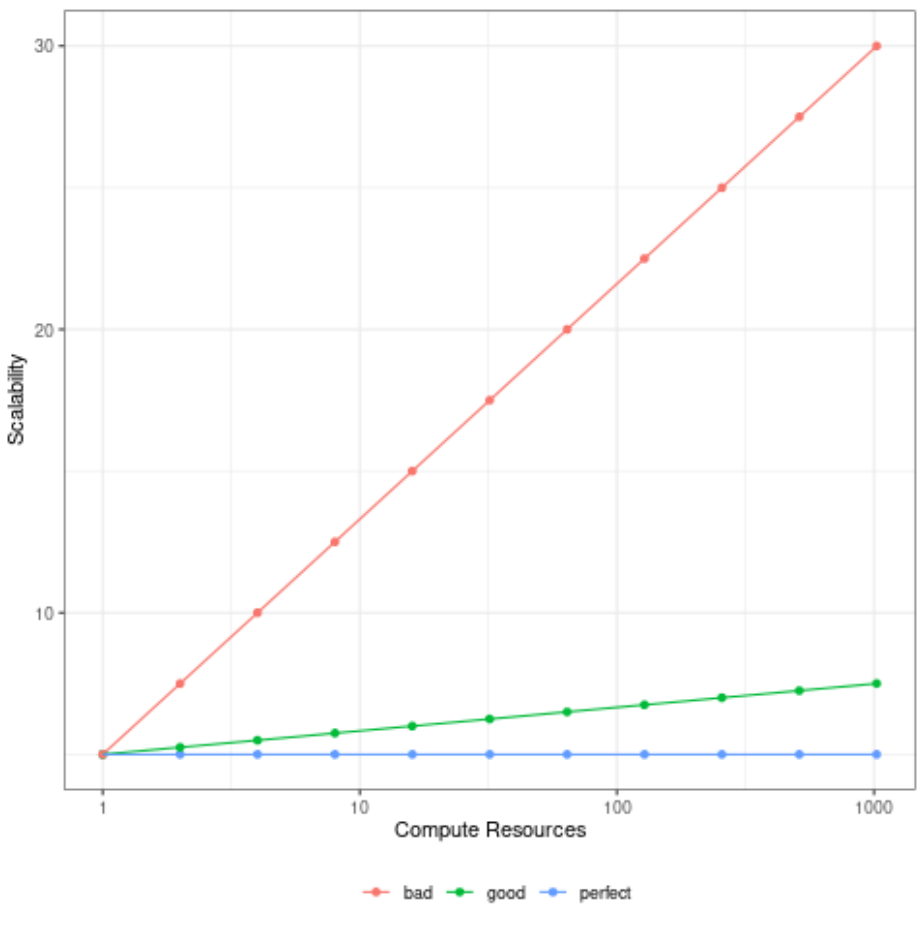


Scalability

Strong Scaling



Weak Scaling

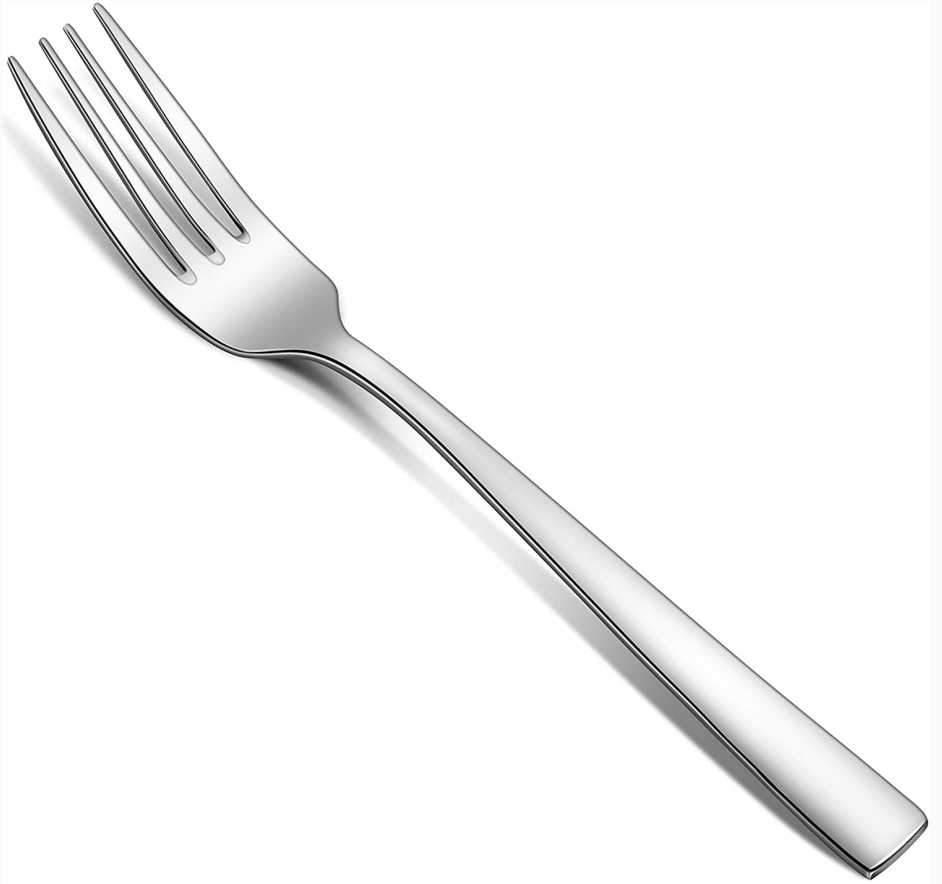


Scalability?

- "scale up"
- "scale out"
- What do these mysterious phrases mean?

Forks and Threads

- Primary interfaces
  - Python's multiprocessing package
  - R's parallel package
- *Many* other interfaces

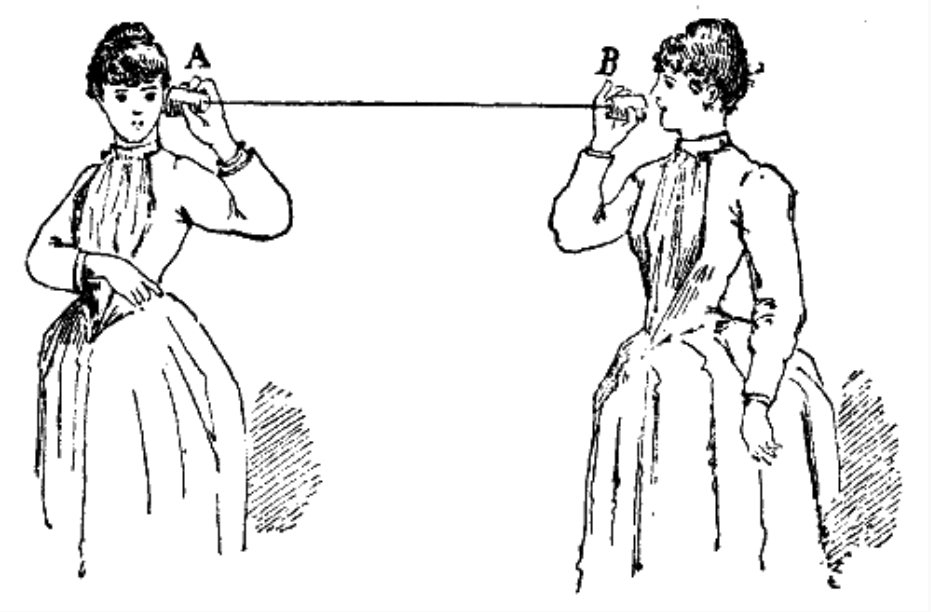


Forks and Threads

- Kernel-level parallelism mechanisms
- Fork
  - Fork is an inexpensive process clone mechanism
  - Only supported on Linux and MacOS
  - Data is copy on modify!
  - Example: have you ever used R/Python paralellism before?
- Threads
  - Sub-component of a process
  - Only *directly* accessible via C/C++ (pthreads, OpenMP, ...)
  - Data is managed by you!
  - Example: OpenBLAS uses threading

MPI

- Message Passing Interface
- Distributed programming standard
- Implementations
  - OpenMPI
  - MPICH
  - MPT
  - Spectrum



HLL Hello World

Python

```
print('Hello World')
```

R

```
print("Hello World")
```

mpi4py Hello World

```
from mpi4py import MPI

comm = MPI.COMM_WORLD
rank = comm.Get_rank()
size = comm.Get_size()

comm_localrank = MPI.Comm.Split_type(comm, MPI.COMM_TYPE_SHARED, 0)
rank_local = comm_localrank.Get_rank()
size_local = comm_localrank.Get_size()

for p in range(0, size):
    if p == rank:
        print("Hello from rank ", end="")
        print(str(rank) + "/" + str(size) + " global ", end="")
        print(str(rank_local) + "/" + str(size_local) + " local")

    comm.Barrier()

mpi4py.MPI.Finalize()
```

pbdR Hello World

```
suppressMessages(library(pbdMPI))

rank = comm.rank()
size = comm.size()
rank_local = comm.localrank()

hostname = system("uname -n", intern=TRUE)
hostnames = allgather(hostname) |> unlist |> table
size_local = hostnames[hostname] |> unname

msg = paste0("Hello from rank ", rank, "/", size, " global ", rank_local, " local\n")
comm.cat(msg, all.rank=TRUE, quiet=TRUE)
```

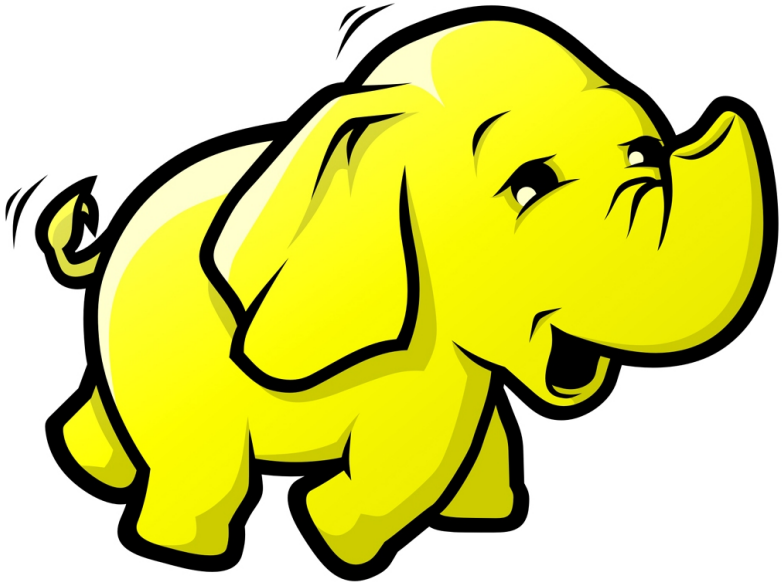
MPI

- Not fork or thread-based --- network-based
- Can use forks or threads within an MPI process (careful with fork…)
- For performance, *you're almost always* better off not mixing MPI + threads
- MPI + GPU on the other hand…

We will have *a lot* to say about MPI…

MapReduce

- The MapReduce algorithm
- Not just `map()` + `reduce()` !
- Implementations
  - Hadoop
  - Spark
  - MPI?!
- Good for ETL; *atrocious* for analytics



Wrapup

- There are many mechanisms for parallel computing.
- We will focus mostly on fork and MPI.
- What does scalability really mean? Was he lying about that weak scalability thing?

Questions?