Introduction to High-Performance Computing (HPC)

Tracy Bu FoA Advanced Tutorial Semester One 2024

What is High-Performance Computing?

- **Definition**: High-Performance Computing involves advanced computation over parallel processing, enabling higher performance than a typical desktop or workstation.
- **Importance**: Essential for complex simulations, data analysis, and scientific research.
- **Applications**: Weather forecasting, AI, Astrophysics, modelling

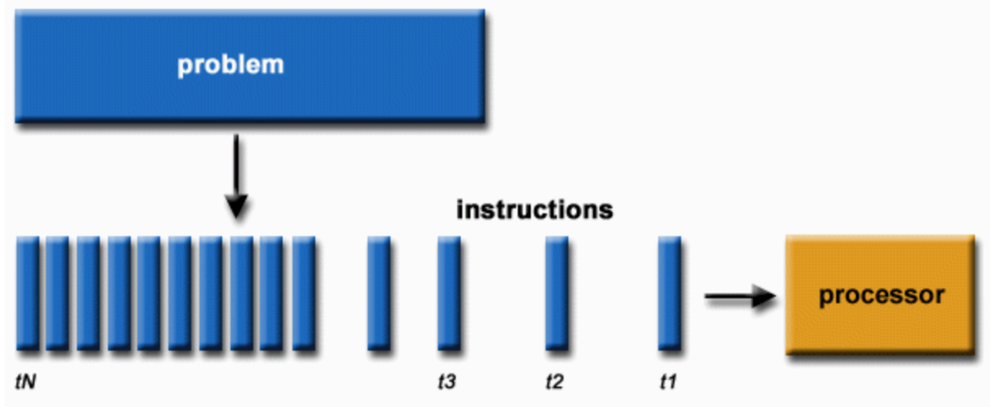
Why do we need HPC?

- Tracy's research needed her to run a lot of simulations
  - Each simulation takes ~1s to run
  - For each configuration, we need to run 10^4 simulations
  - For my results to be statistically reliable, I need to test on 100 configurations
  - That is $1 \times 10^4 \times 100 = 10^6 \text{s} \simeq 178\ \text{h} \simeq 12\ \text{days}$
  - We need to do it faster
- It would be nice if could run each configuration at the same time, or, in **parallel**

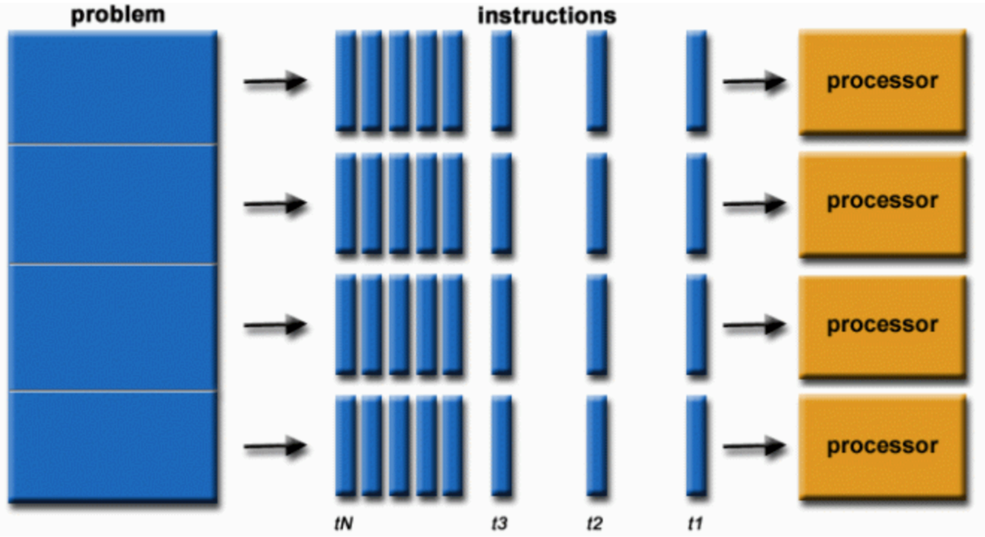
Serial vs. Parallel Programming

Kinda like electric circuits

- **Serial Programming**: Executes one instruction at a time. Simpler but slower for complex calculations.



- **Parallel Programming**: Divides tasks into smaller ones that can be processed simultaneously.
- Benefits: Reduces processing time significantly.

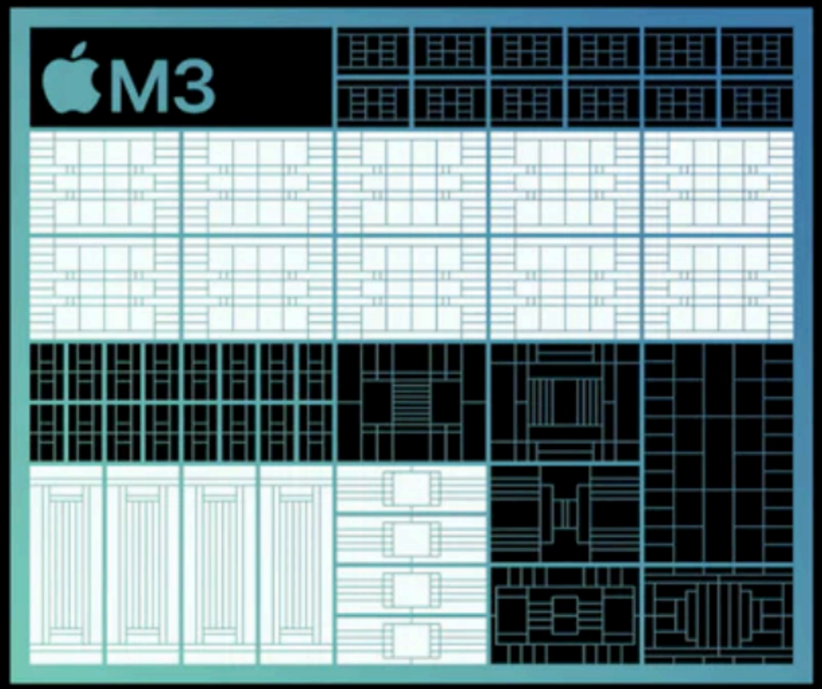


- Example — is_prime(int n)

https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial##SPMD-MPMD

CPU

Up to 24GB of unified memory
25 billion transistors



8-core CPU

4 performance cores
4 efficiency cores
Up to 35% faster than M1
Up to 20% faster than M2

10-core GPU

Next-generation architecture
Dynamic Caching
Mesh shading
Ray tracing
Up to 65% faster than M1
Up to 20% faster than M2

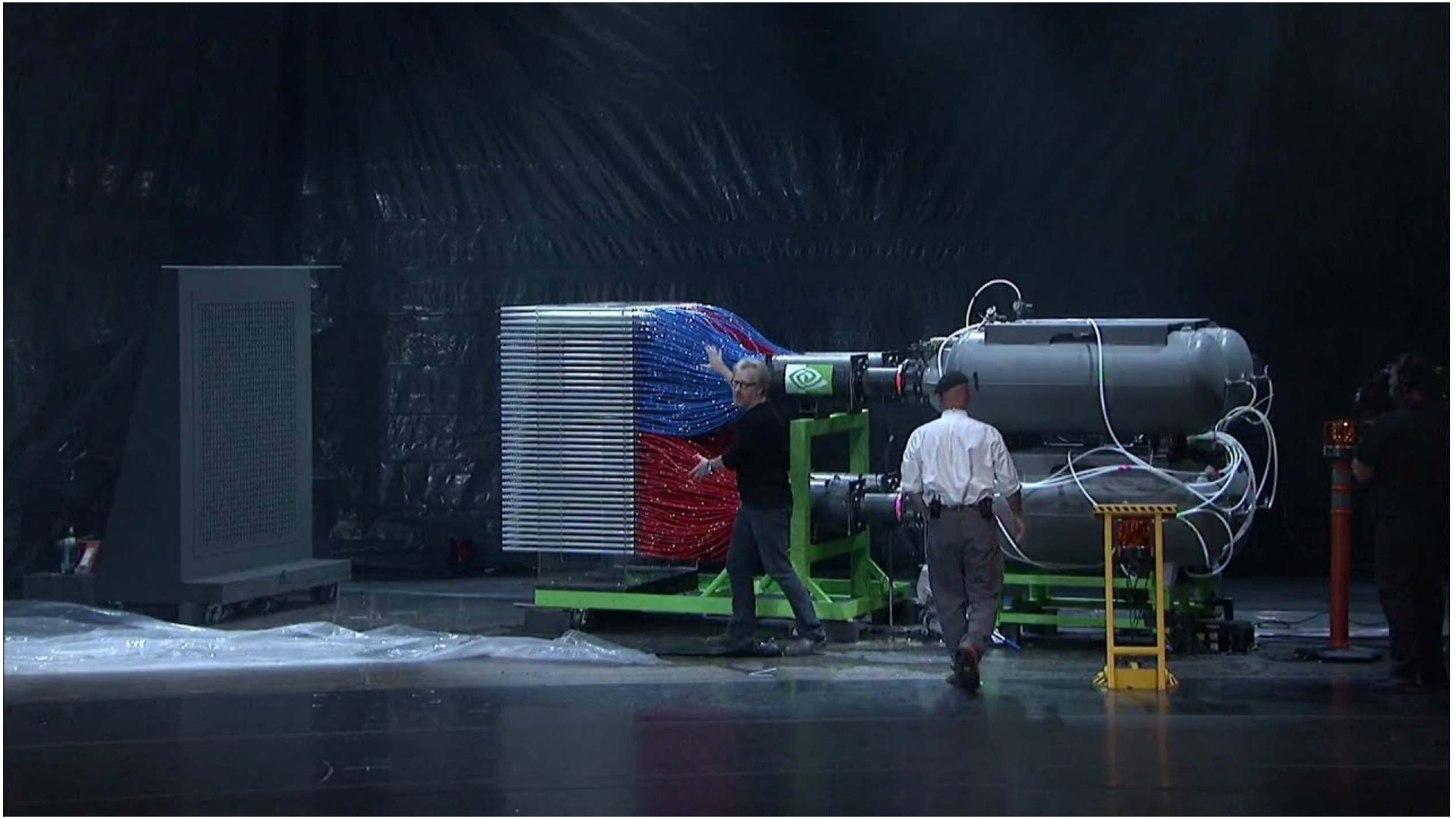
CPU

On a very high-level

- Multi-core CPU — A team
- Core — A worker
- A thread — one task
- OS — The manager
- The manager delegate task to workers, a team work on many tasks at the same time
- NB: The word thread is overloaded, can mean different things

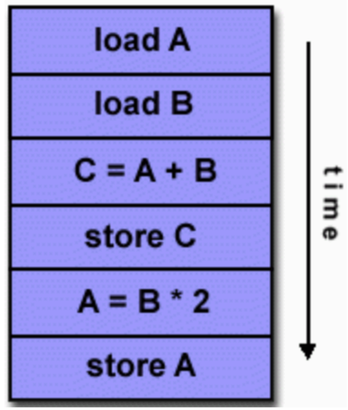
Parallel Computing Architectures

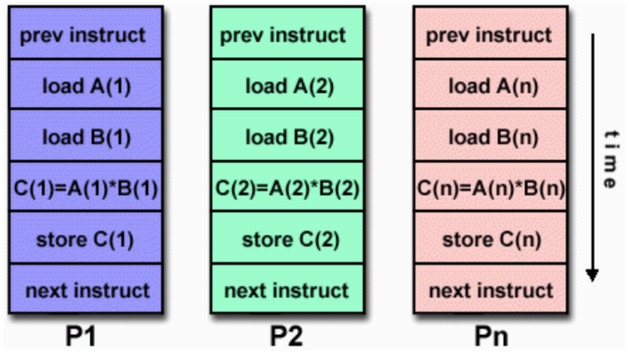
- Shared Memory: Multiple processors accessing the same memory space
- Challenges: Synchronization, avoiding race conditions.
- Distributed Memory: Each processor has its own memory; processors communicate via messages.
- e.g. GPU

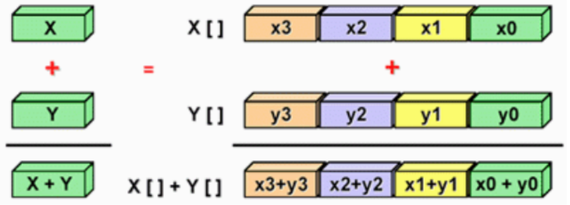


CPUs and GPUs

- CPU: Central processor suitable for **general-purpose tasks**, handles diverse tasks simultaneously.
  - Good at many types task, slower
- GPU: **Specialized** for simultaneous processing of multiple tasks, significantly faster for algorithms that process large blocks of data simultaneously.
  - Good at a specific type of task; graphics!







GPUs

- SIMD — single instruction multiple data
  - Each processing unit is doing the same thing at the same time
  - But on a different data element
    - e.g. Matrix add : All adding, but adding different elements
- Usually have a lot more cores than CPUs
  - Apple M3 - 8 cores
  - RTX4090 - 16,384 cores
- CUDA — Can use GPU for more general purposes

100 SECONDS OF

CUDA

What are GPU used for

- Games, graphics rendering
- REALLY big scientific compution — Gravitational wave search
- Machine learning
- Large Language models — ChatGPT!

Even more powerful — Supercomputer

A lot a lot of computers in a network

- Frontier — most powerful — 8 million cores
- OzStar — 482,752 cores

Programming Models for Parallel Computing

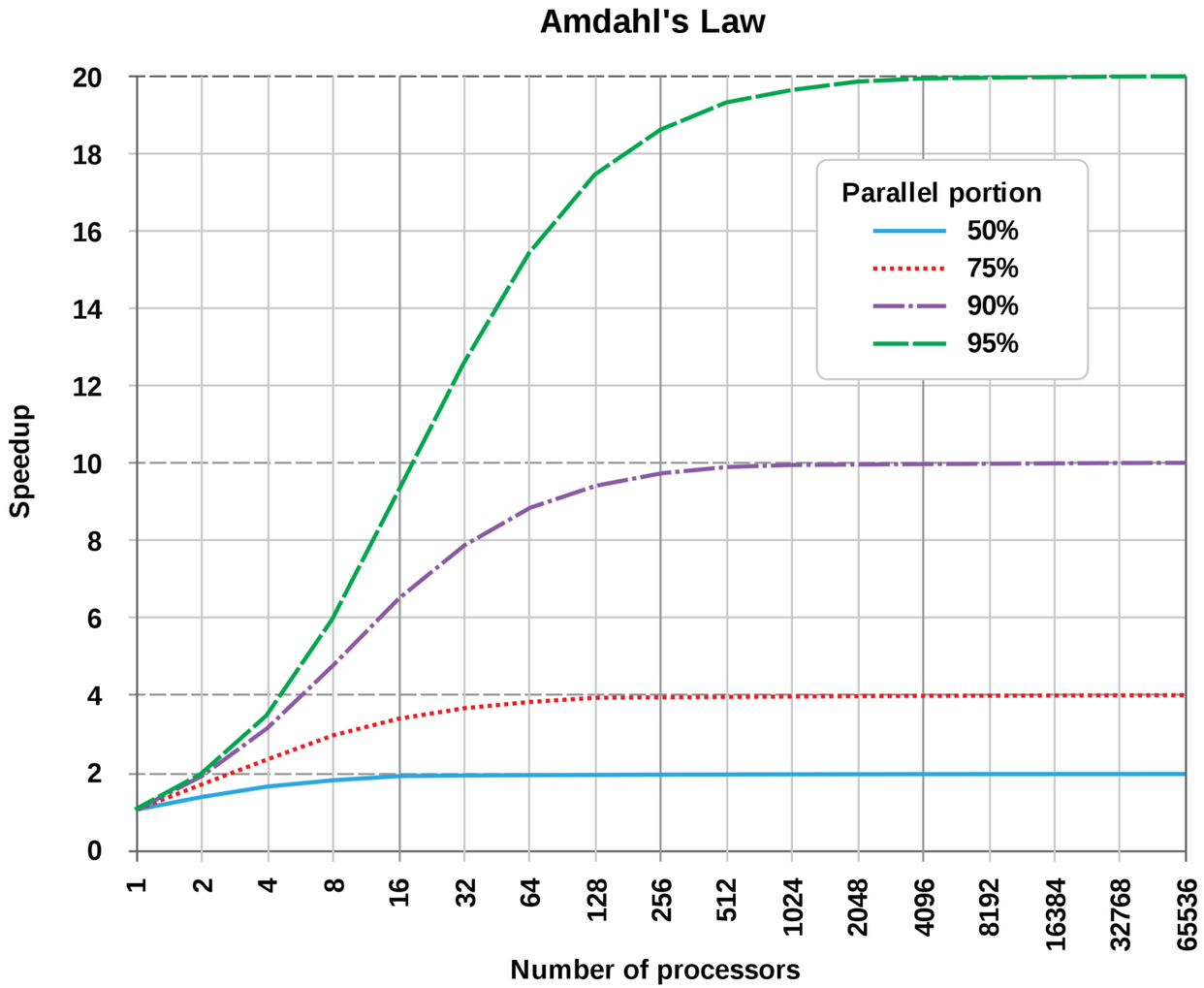
Communication is key

- Which core to do what?
- Where to store the result?

- OpenMP: A model for multi-threaded programming
- MPI (Message Passing Interface): Allows communication between nodes in a distributed system

Amdahl's Law

More is not better



Future Trends in HPC

- Technological Advances: Quantum computing(NEXT WEEK!), specialized AI chips.
- Software Developments: New frameworks and libraries for more efficient parallel programming.