CSCI 325
Introduction to Parallel Systems and GPU Programming

Lecture 1
Parallel software and hardware organization

Dr. Talgat Turanbekuly

Table of contents

Definitions

Cluster - a set of computers connected over a local area network (LAN) that functions as a single large multiprocessor.

Parallel processing program - a single program that runs on multiple processors simultaneously.

Job level parallelism - utilizing multiple processors by running independent programs simultaneously.

Multiprocessor - a computer system with at least two processors.

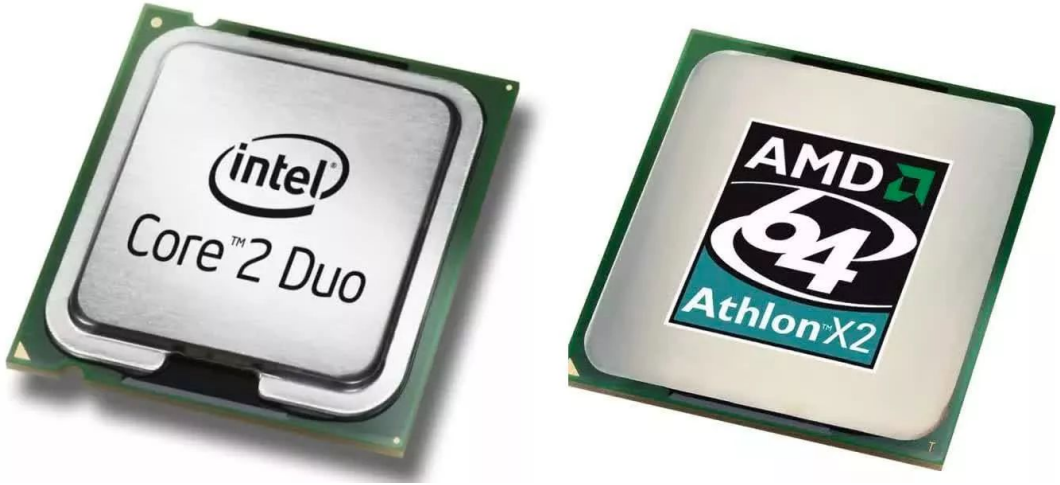
Multicore, Multiprocessor - a microprocessor containing multiple processors (“cores”) in a single integrated circuit.

HARDWARE

https://c4.wallpaperflare.com/wallpaper/188/76/864/amd-nvidia-intel-asus-wallpaper-preview.jpg

Multicore Processor

“Multiprocessor - a computer system with at least two processors.”

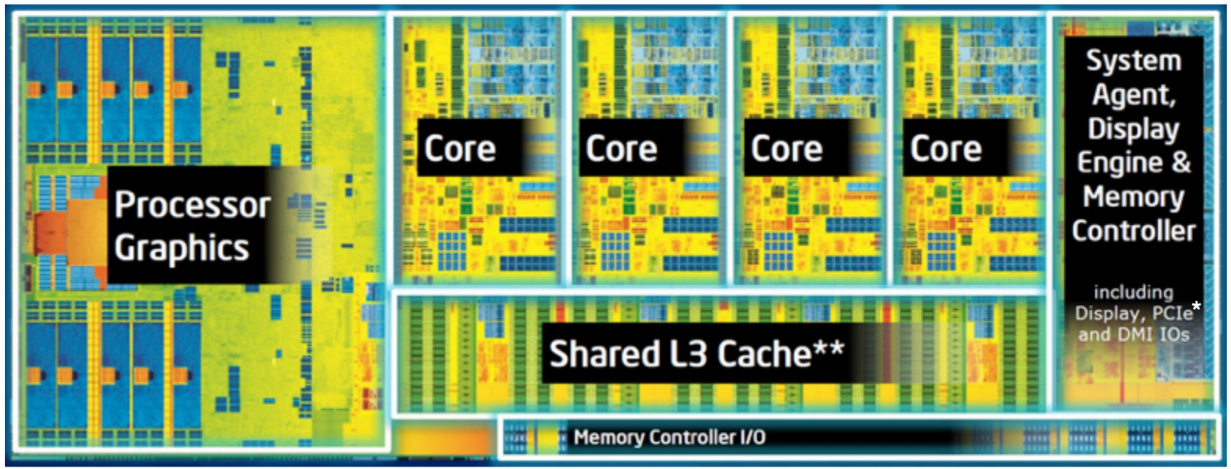


Patterson, David A, and John L Hennessy. *Computer Organization and Design: The Hardware/Software Interface*. 4th ed. San Diego: Elsevier Science & Technology, 2011. Print.

https://www.techspot.com/article/2363-multi-core-cpu/

Multicore Processor

“Multicore, Multiprocessor - a microprocessor containing multiple processors (“cores”) in a single integrated circuit.”



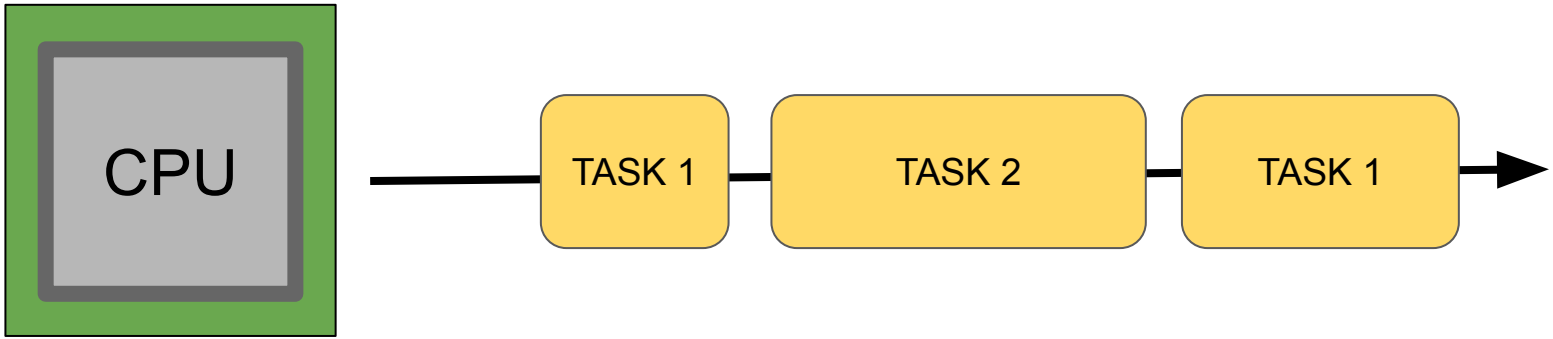
Patterson, David A, and John L Hennessy. *Computer Organization and Design: The Hardware/Software Interface*. 4th ed. San Diego: Elsevier Science & Technology, 2011. Print.

https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/ia-introduction-basics-paper.pdf

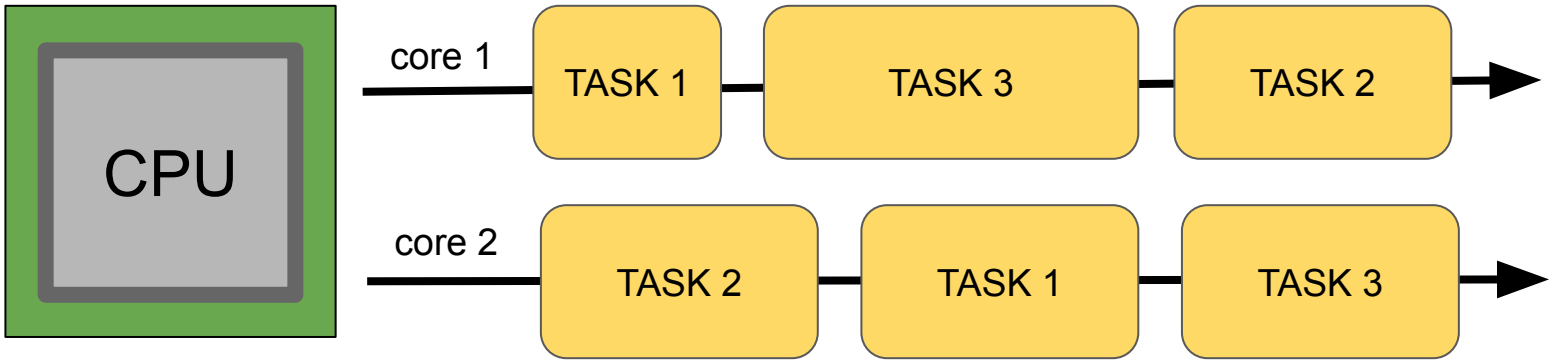
Concurrency

https://i.pcmag.com/imagery/articles/07byLBF5UaGsFsKtyOsENAg-16..v1643148022.png

Concurrency

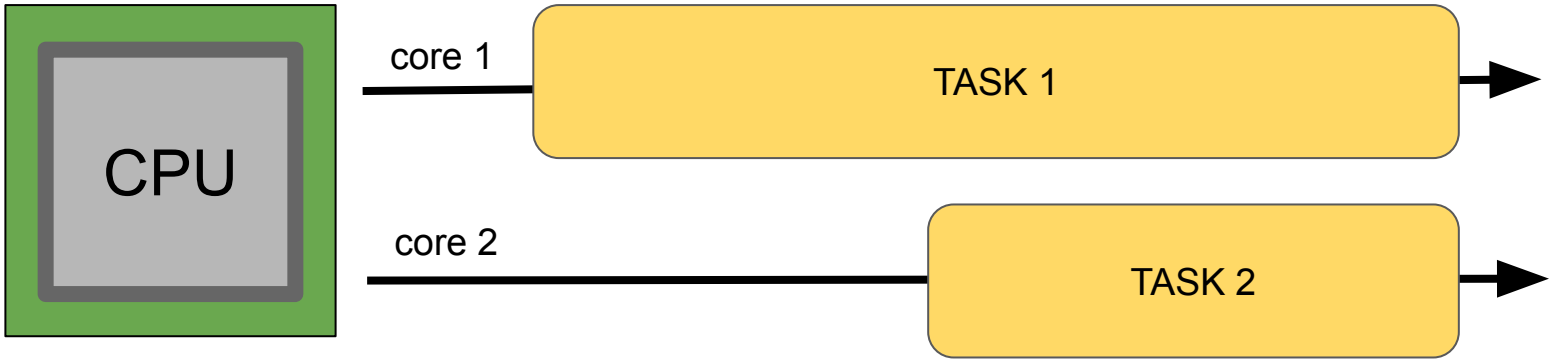


Concurrency



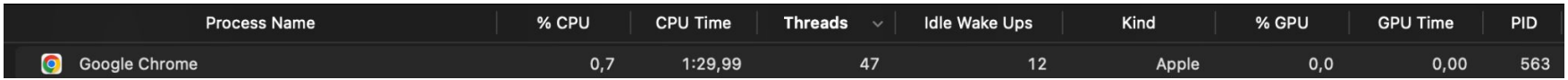
Parallel Programming

“Job level parallelism - utilizing multiple processors by running independent programs simultaneously.”



Patterson, David A, and John L Hennessy. *Computer Organization and Design: The Hardware/Software Interface*. 4th ed. San Diego: Elsevier Science & Technology, 2011. Print.

Process, Thread



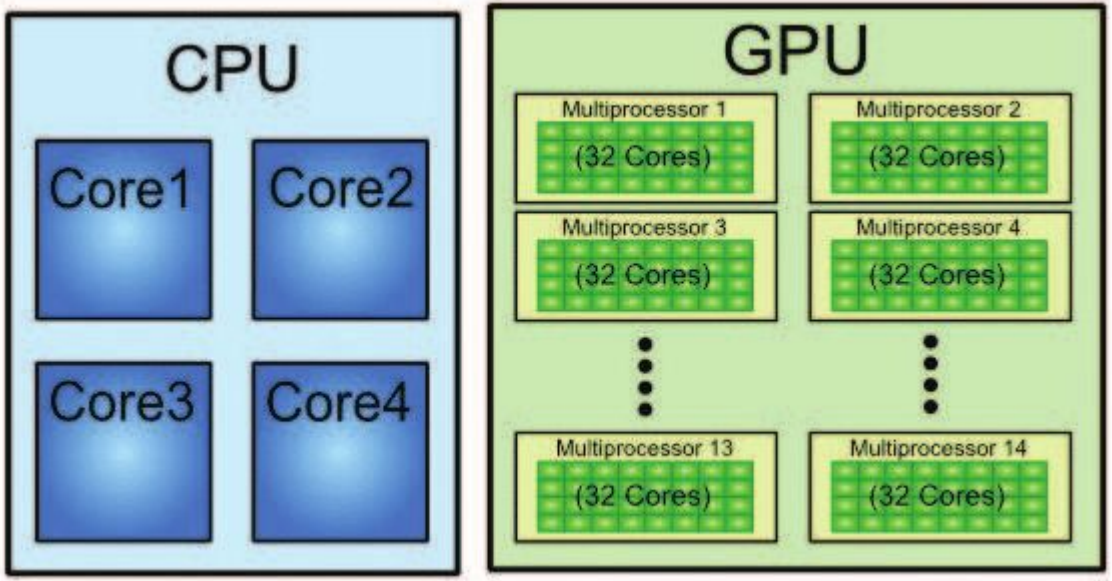
Process Name	% CPU	CPU Time	Threads	Idle Wake Ups	Kind	% GPU	GPU Time	PID
Google Chrome	0,7	1:29,99	47	12	Apple	0,0	0,00	563

Process - running program with its own space.

Thread - a sequence of executable instructions within a process.

In the screenshot above, 47 threads are assigned for process named “Google Chrome”.

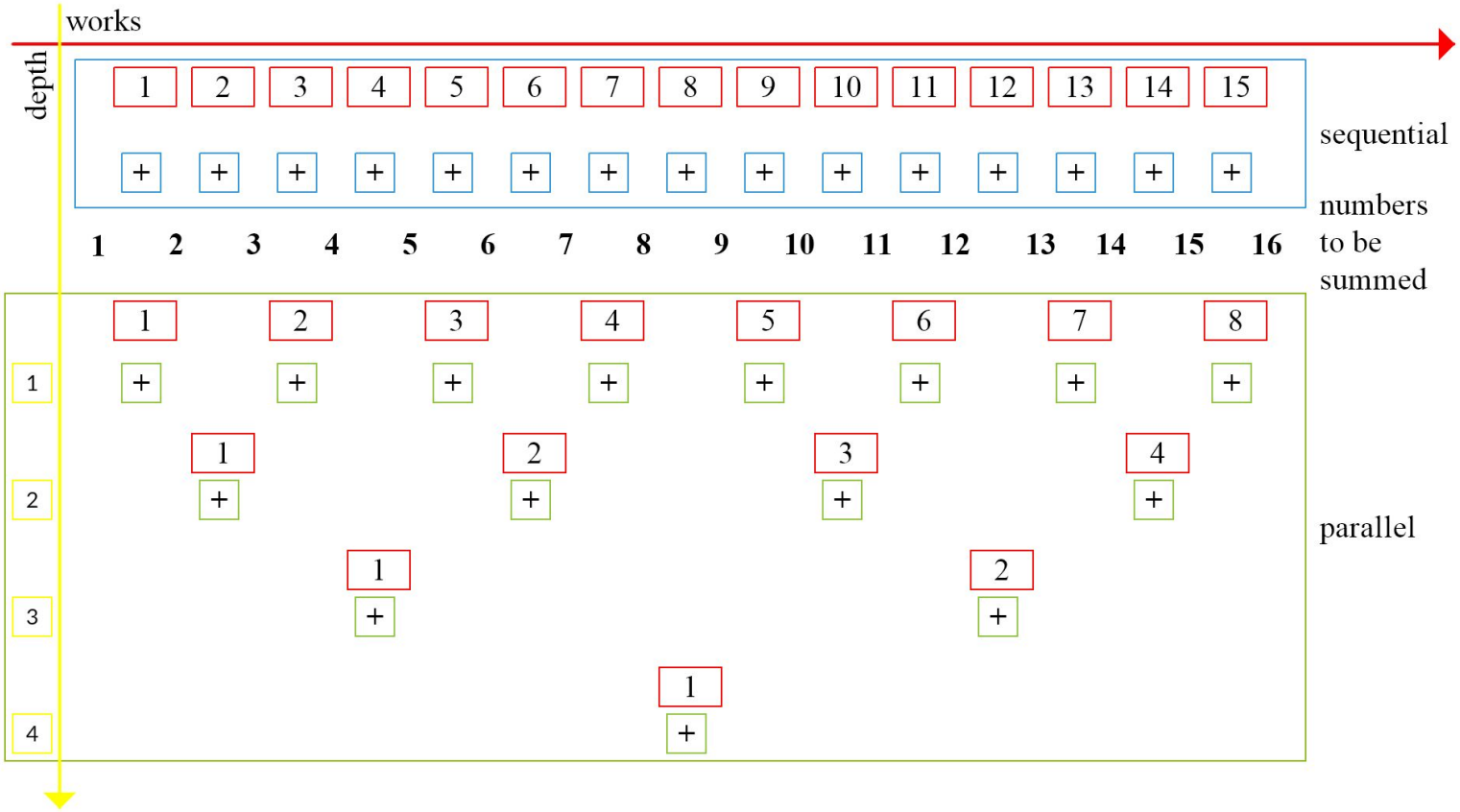
Multicore GPU architecture



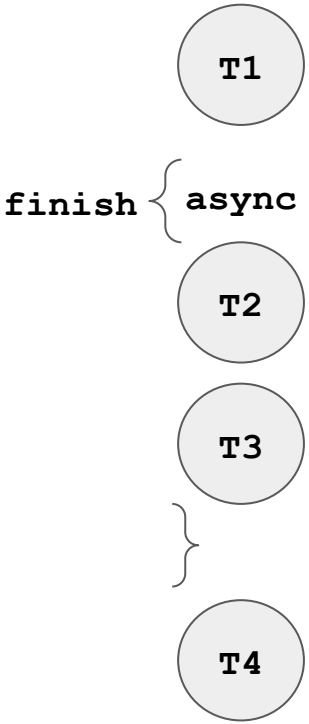
https://www.researchgate.net/publication/319468789_Mobile_devices'_GPUs_in_cloth_dynamics_simulation

https://getwallpapers.com/wallpaper/full/c/5/d/688482-large-software-wallpapers-1920x1080-hd-1080p.jpg

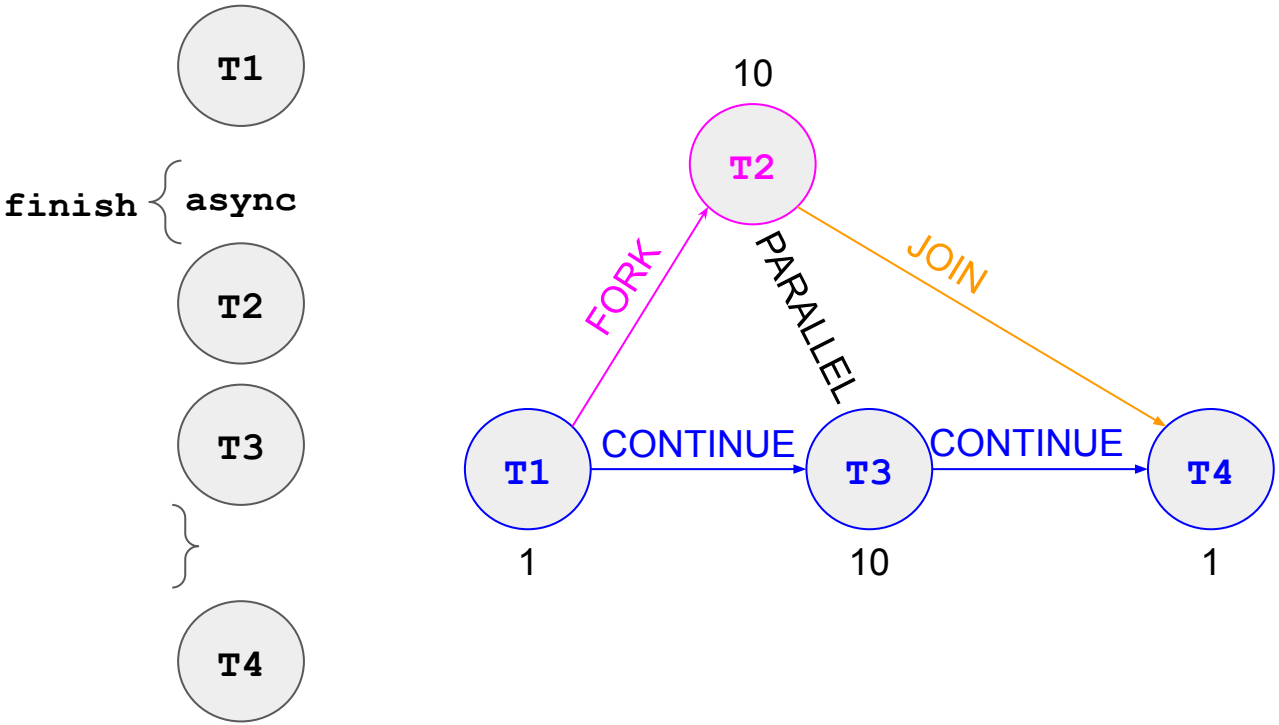
Parallel Programming Work and Span (Depth)



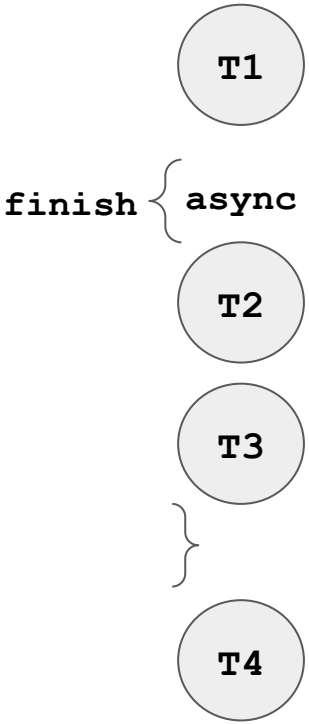
Multiprocessor Scheduling, Parallel Speedup

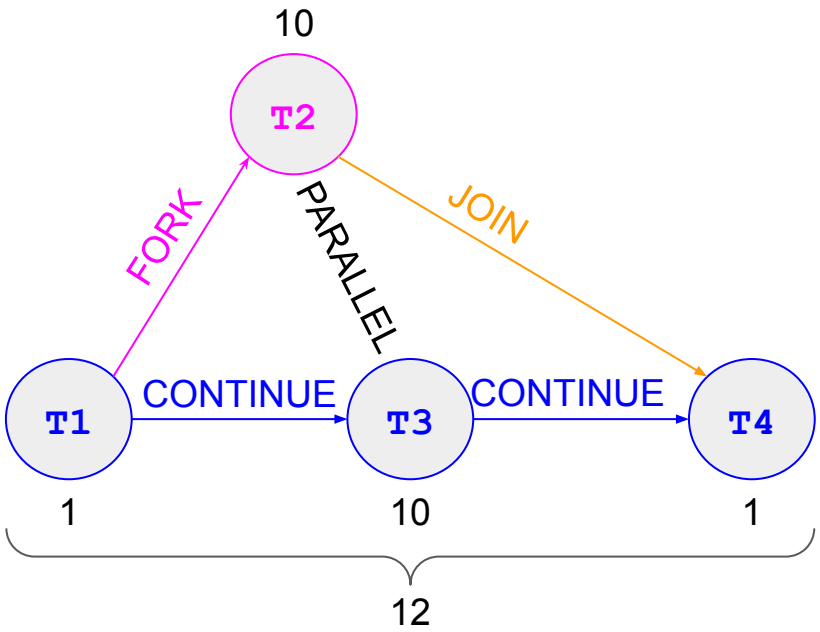


Multiprocessor Scheduling, Parallel Speedup



Multiprocessor Scheduling, Parallel Speedup





work = 1 + 10 + 10 + 1 = 22

span = 12

ideal parallelism <= $\frac{\text{work}}{\text{span}}$

Amdahl’s Law

Q - sequential part of the program

$$\text{SPEEDUP} <= \frac{1}{Q}$$

Amdahl’s Law

Q - sequential part of the program

$$\text{SPEEDUP} <= \frac{1}{Q}$$

example 1

the portion of the program that runs only sequentially - 50 % or 0.5

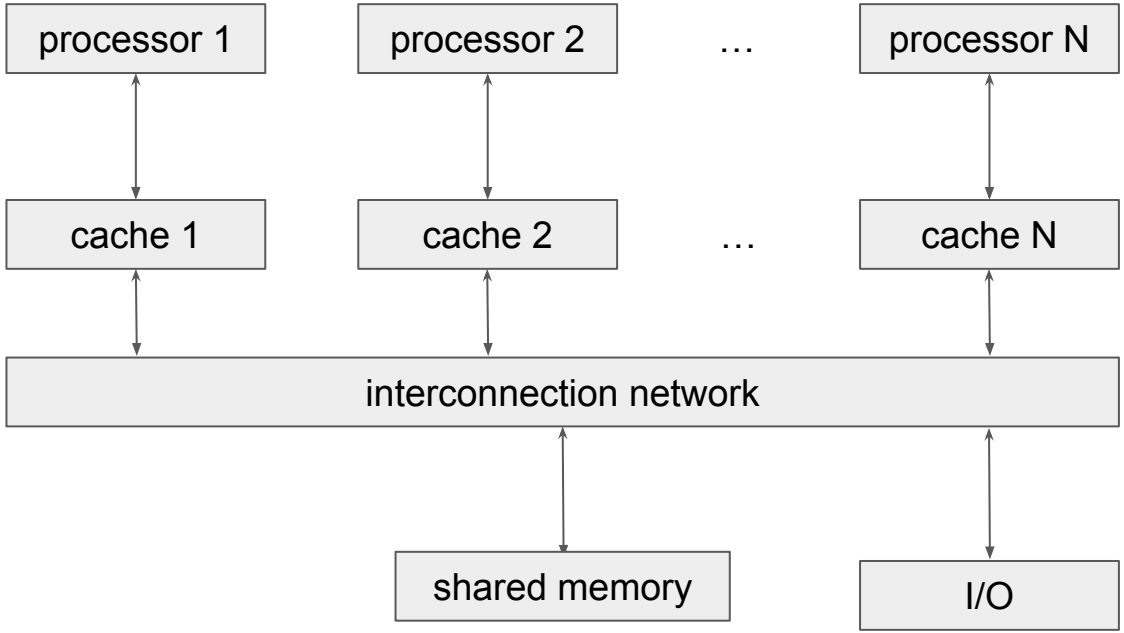
maximum speedup 1 / 0.5 = 2;

example 2

the portion of the program that runs only sequentially - 10 % or 0.1

maximum speedup 1 / 0.1 = 10;

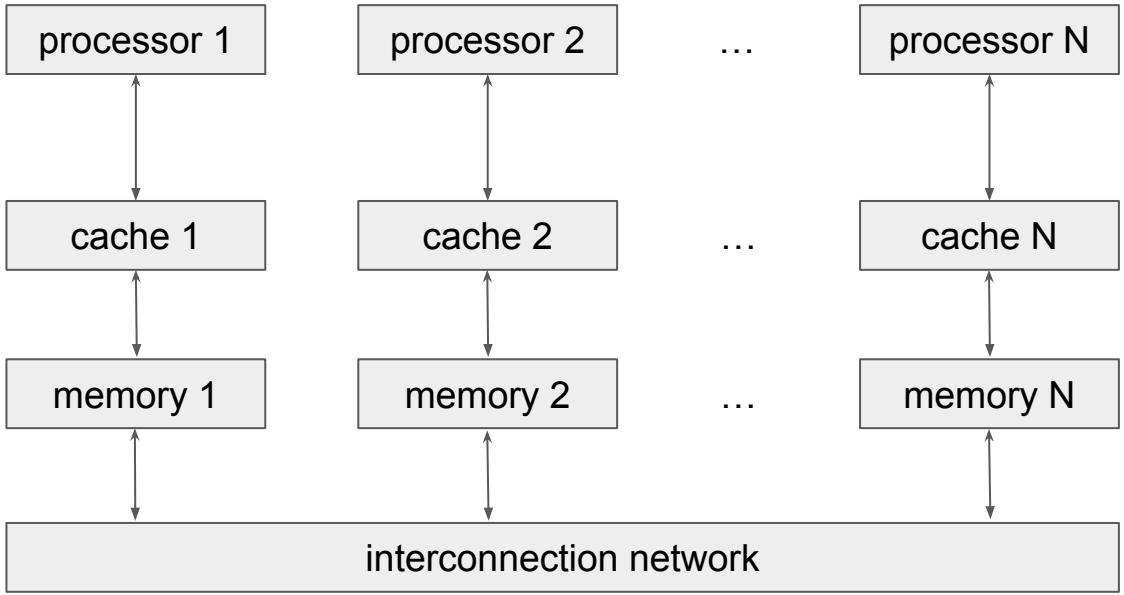
Shared Memory Multiprocessor



synchronization - *coordination of multiple processors when operate on shared memory.*

administration cost for N processors same as for single machine.

Message Passing Multiprocessor



message passing - *sending and receiving messages among processors from their private memories.*

High communication performance, but expensive.

Clusters vs. Virtual Machines

Clusters - computers connected via standard network using switches and cables.

administration as costly as the size;

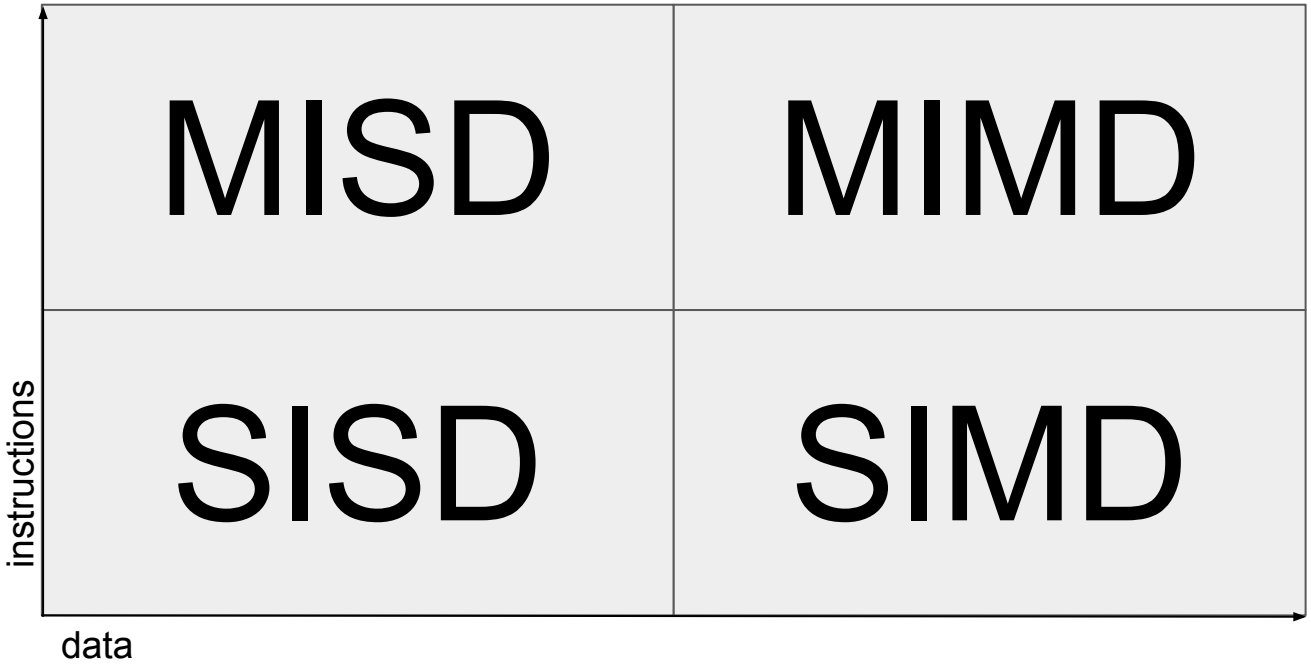
easy to increase/decrease the size;

Virtual machines - operating systems run within other OS on a physical machine

start/stop programs independently;

migrate running program within computers;

Instructions and Data



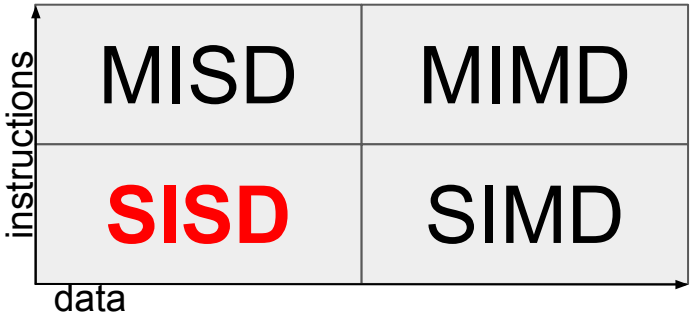
S - SINGLE

M - MULTIPLE

I - INSTRUCTION

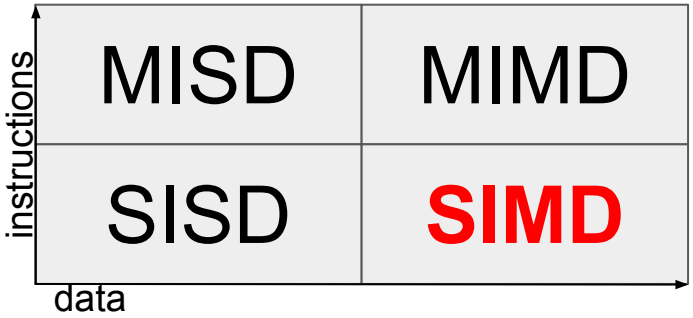
D - DATA

Instructions and Data - **SISD**



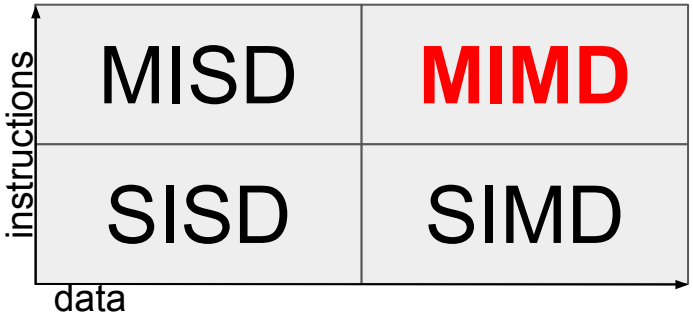
- common personal computers with one instruction per one stream of data;
- limited performance by processor;
- optimization using concurrency and pipeline;

Instructions and Data - **SIMD**



- the same instruction (portion of the program) run on multiple data;
- multiple data must be identically structured (arrays);
- each instruction execution unit has its unique address register;

Instructions and Data - **MIMD**



- separate instructions run on different processors;
- different processors work on different parts of a program;

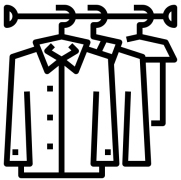
Pipeline (example with laundry)

1. Put dirty clothes to the laundry

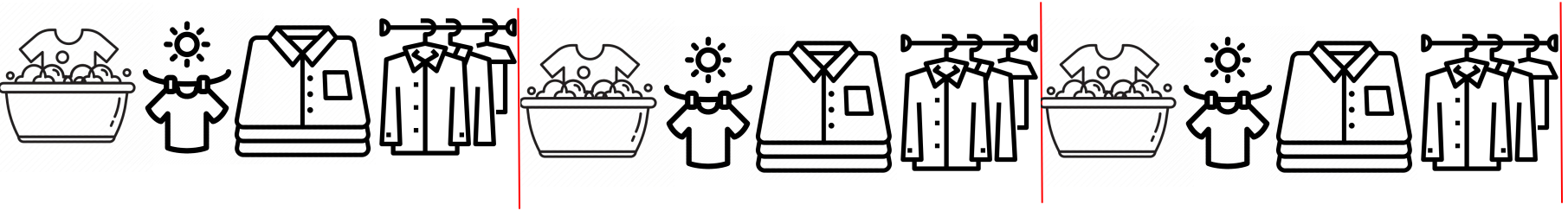
2. Put washed clothes to the dryer

3. Fold dried clothes

4. Put clothes away

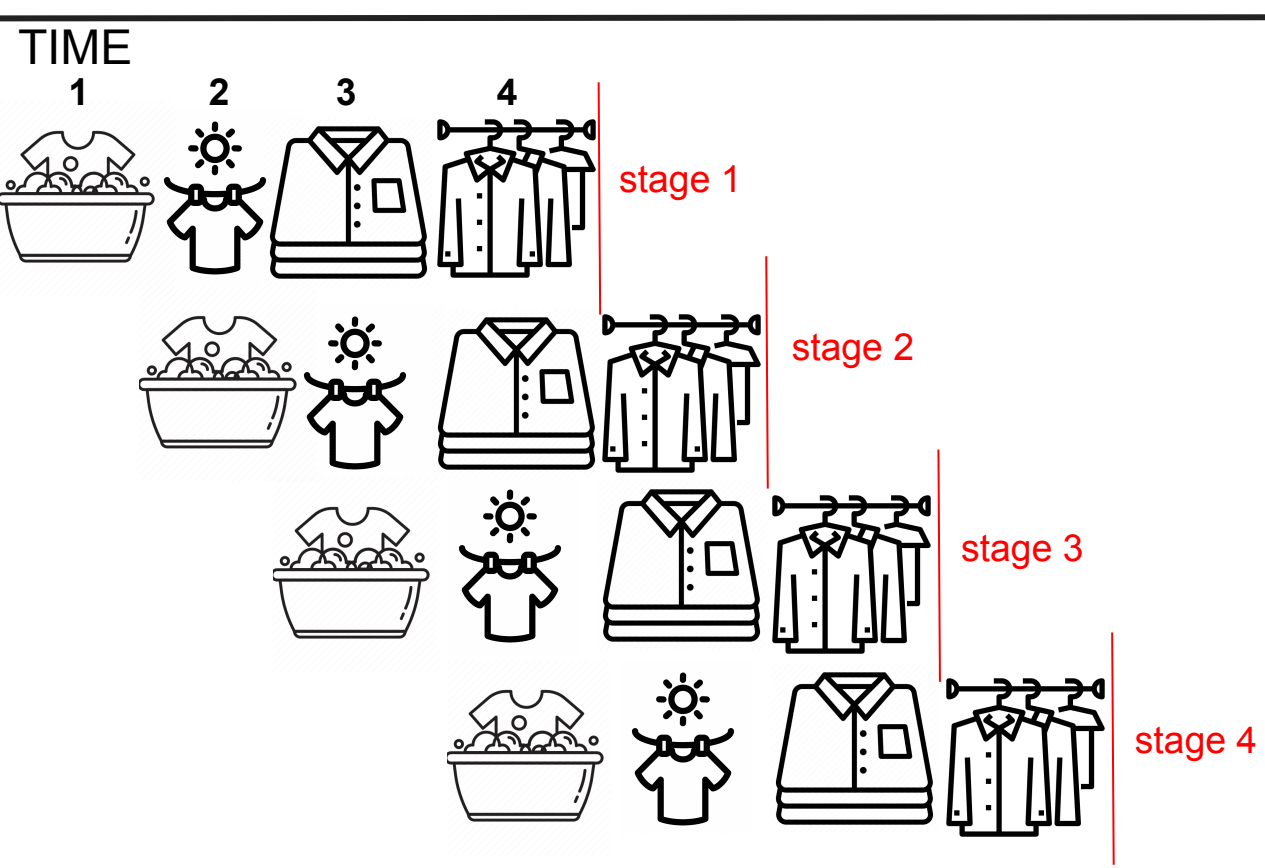


Pipeline (example with laundry)



TIME

Pipeline (example with laundry)

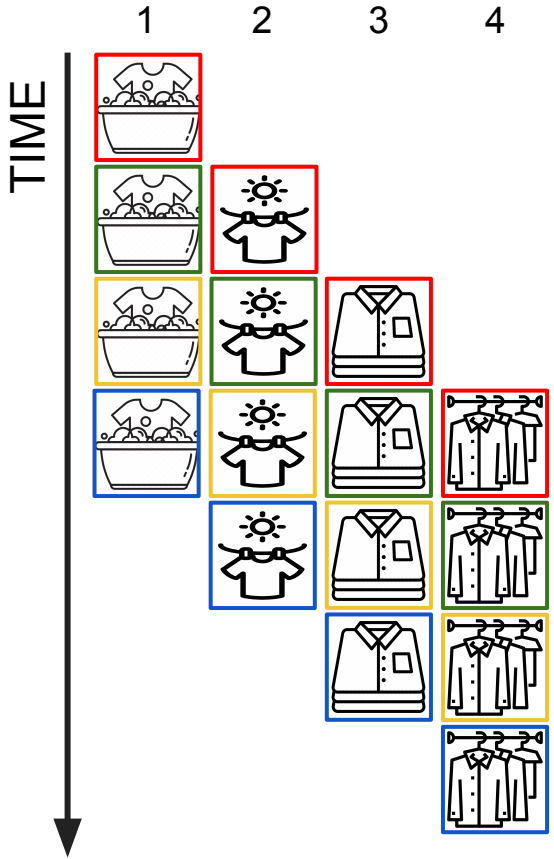


work = 4 x 4;
CPL = 4 + (4 - 1);
Parallelism = (4x4)/(4+4-1) ≈ 2.3;

work = N x P;
CPL = N + (P - 1);
Parallelism = (N*P) / (N + P - 1)

N - work in each stage;
P - pipeline stages;
CPL - critical path length;

Pipeline (example with laundry)



- Increases the number of simultaneously executing instructions;
- Does not reduce the time to complete individual instructions;

C++ Multithreading

Example

C++ Multithreading

1. Declare an array or vector of integers with 100000 elements. Fill it with ones. Launch 10 threads and break the array equally within the threads. Each thread needs to find sum of all elements within the portion. Finally, sum all the results.

Summary

Parallel programming not always faster

Acceleration requires hardware and effectively developed software

There are lots of techniques for parallelization and software optimization

There are existing frameworks and libraries for parallel programming

Resources

- https://www.intel.com/pressroom/archive/releases/2005/20050418comp.htm
- *Computer Organization and Design - The Hardware/Software Interface 4th Edition (Ch. 7)* by Patterson, David A, and John L Hennessy. San Diego: Elsevier, 2009. Print.
- Computer Architecture: A Quantitative Approach 5th Edition (Ch. 3 and Ch. 4) by John L. Hennessy and David A. Patterson, MK Publications.
- Kalin, Martin. *Concurrent and Parallel Programming Concepts*. online O'Reilly, 2015. Print.
- Harvey Deitel, and Paul J. Deitel. C++20 for Programmers: An Object's-Natural Approach, 3rd Edition, 2022;
- Paul J. Deitel. C++20 Fundamentals, 3rd Edition. 2024;
- Anthony Williams. C++ Concurrency in Action, Second Edition, 2019;
- Bartosz Milewski, Introduction to C++ Concurrency LiveLessons (Video Training), 2014
- https://dl.acm.org/doi/pdf/10.1145/1465482.1465560
- https://www.geeksforgeeks.org/computer-organization-amdahls-law-and-its-proof/amp/
- https://dl.acm.org/doi/pdf/10.1145/227234.227246
- https://docs.oracle.com/javase/8/docs/api/java/util/stream/Stream.html