HPI Hasso Plattner Institut
Digital Engineering · Universität Potsdam

Parallel Programming and Heterogeneous Computing

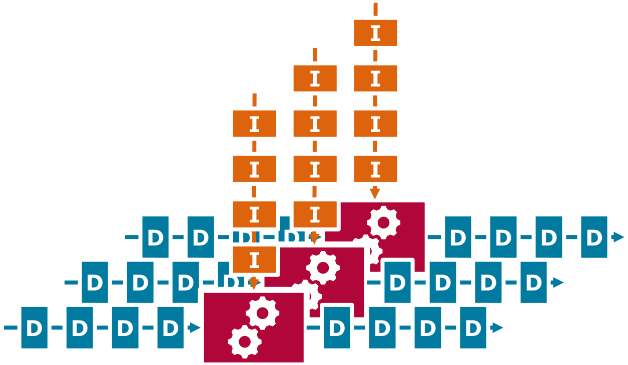
A2 - Parallel Hardware

Max Plauth, Sven Köhler, Felix Eberhardt, Lukas Wenzel and Andreas Polze
Operating Systems and Middleware Group

Types of Parallel Hardware

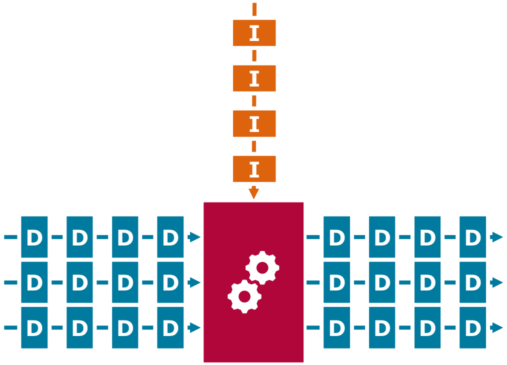
Task Level Parallelism

Multiple operations are executed in parallel.

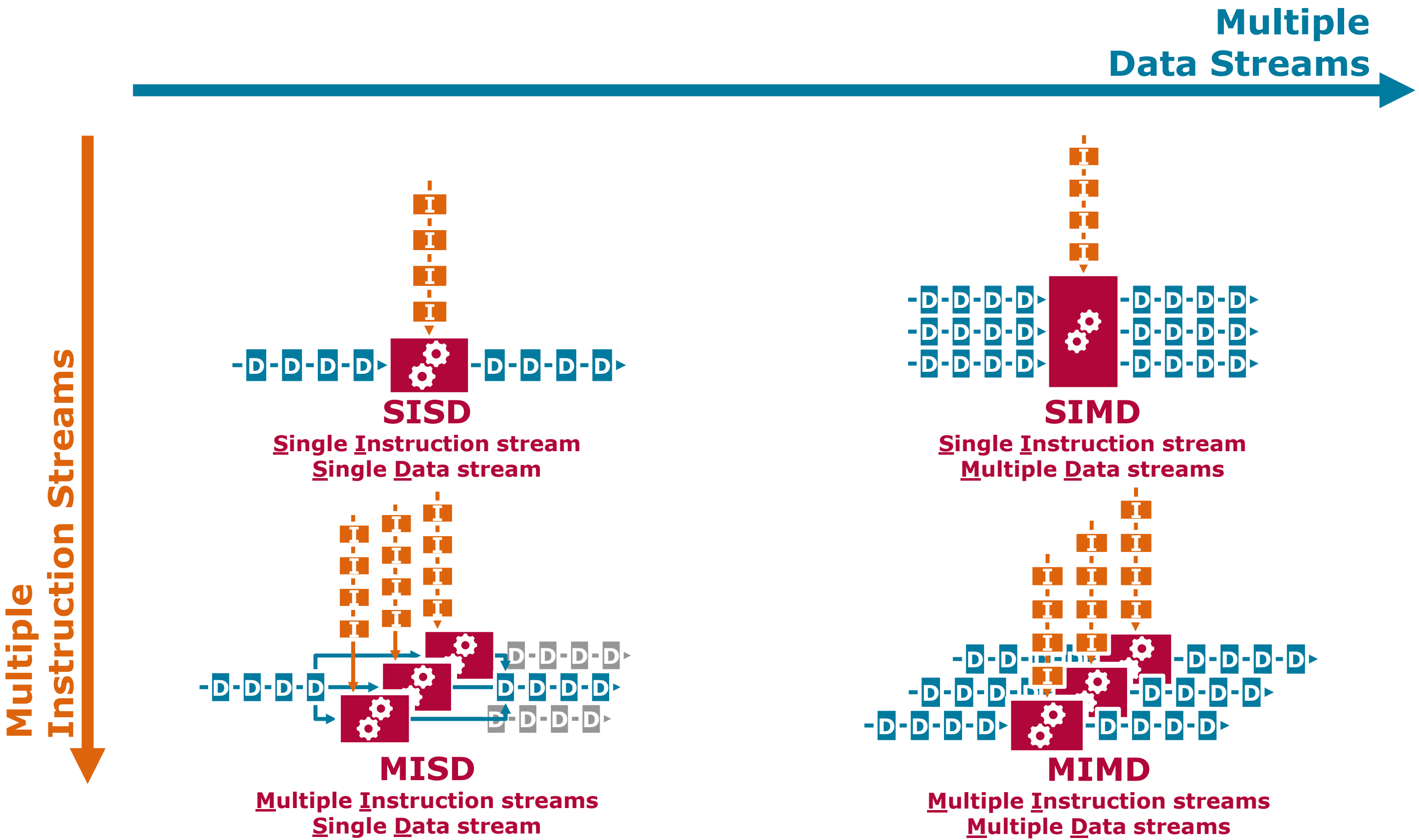


Data Level Parallelism

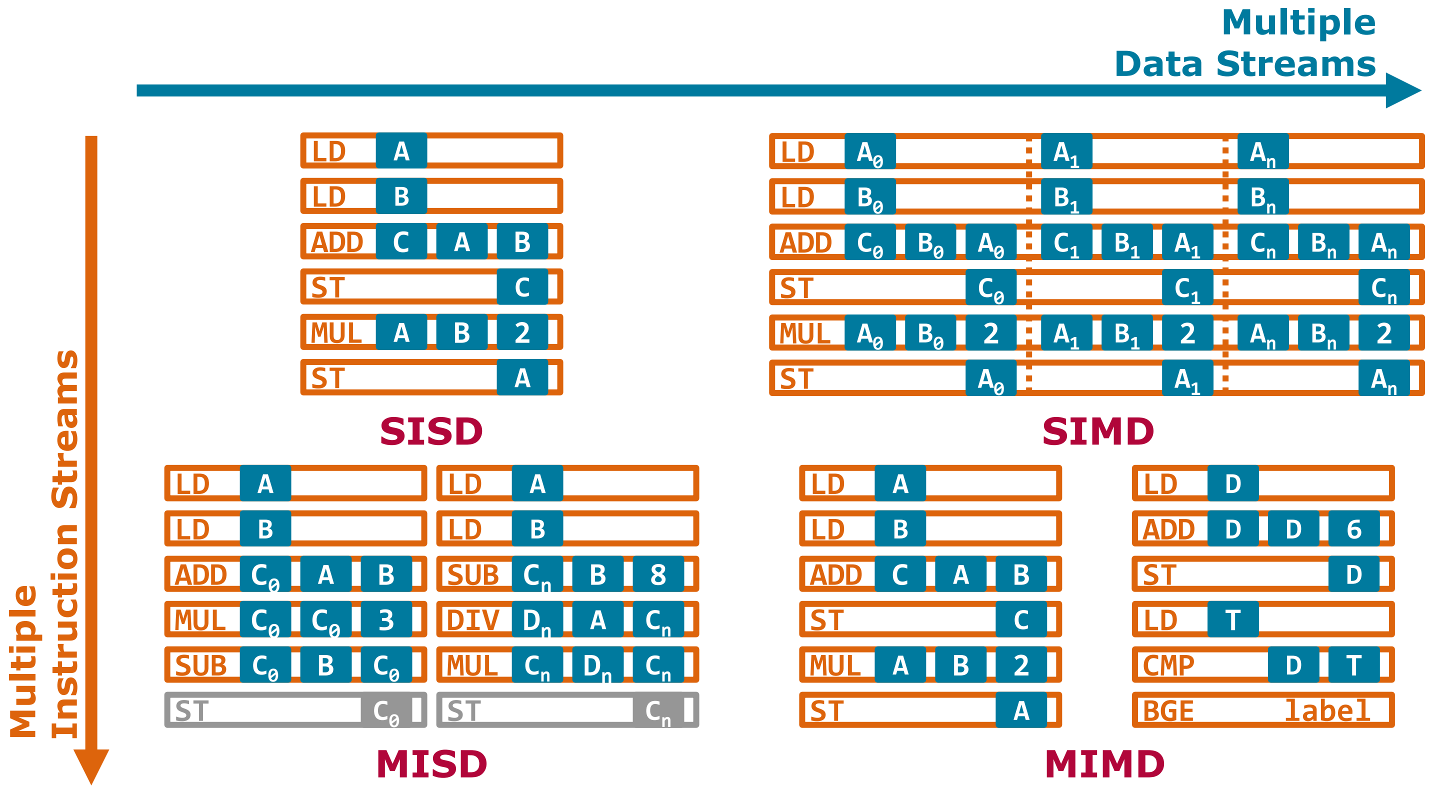
The same operation is applied in parallel to multiple units of data.



Hardware Taxonomy [Flynn1966]



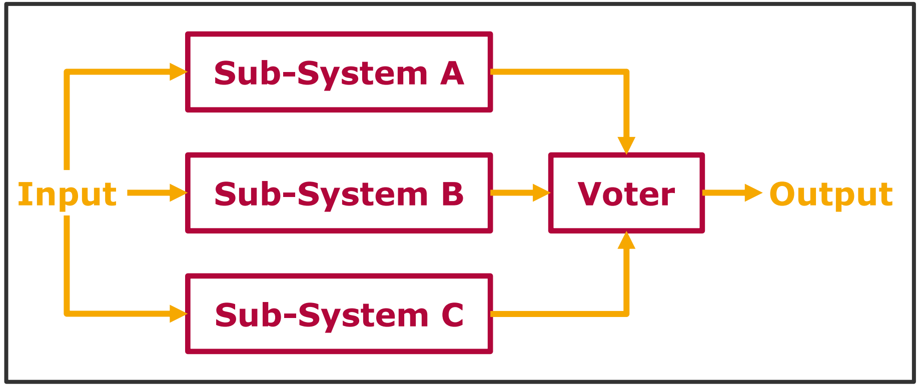
Hardware Taxonomy [Flynn1966]



MISD Hardware

Most exotic class of parallel hardware, not in mainstream use.

- **Redundant systems** like safety-critical embedded controllers or high-reliability mainframes
- Parallelism not for performance, but dependability



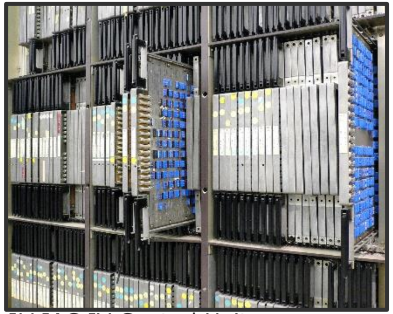
Example: Triple Modular Redundant Architecture

Not covered in this lecture.

SIMD Hardware

Popular class of parallel hardware for special purpose systems.

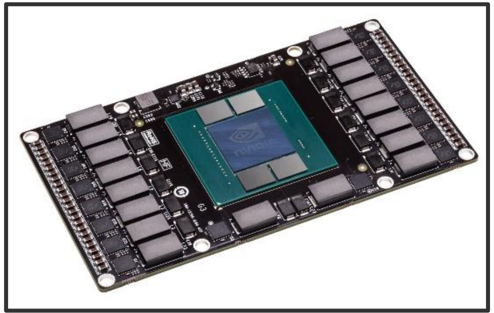
- **Vector processors**
- Early examples: ILLIAC IV, Cray-1, ...


ILLIAC IV Control Unit


Cray-1

Recently in widespread use:

- GPUs
- Instruction Set Extensions (AltiVec, SSE, AVX, ...)

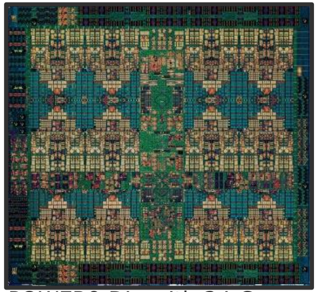

NVidia Pascal GPU Module

Covered in chapter C.

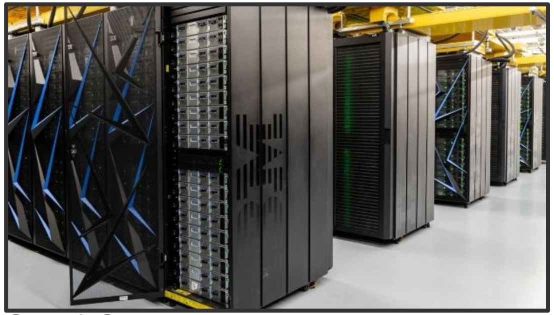
MIMD Hardware

Classic and most general class of parallel hardware.

= **Wide range** of systems from
Multicore CPUs to Supercomputers and Clusters


POWER9 Die with 24 Cores

➢ Variety of architectures and characteristics
requires further distinction

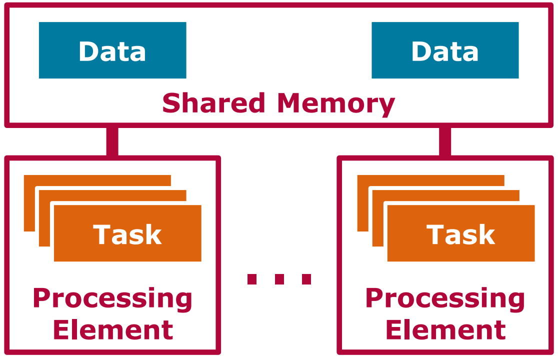

Summit Supercomputer

MIMD Hardware Taxonomy

MIMD

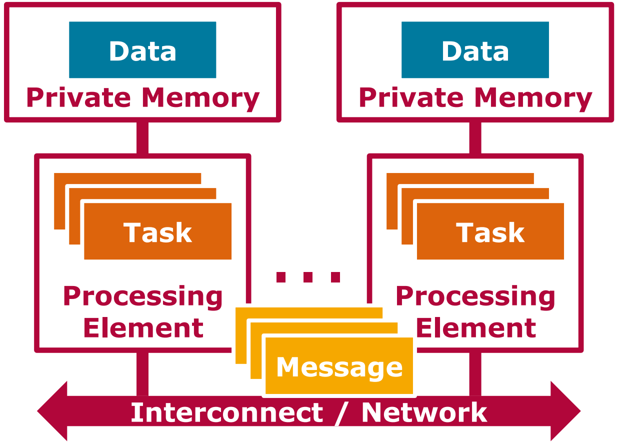
SM-MIMD
(Shared Memory)

Processing elements can directly access a **common address space**



DM-MIMD
(Distributed Memory)

Processing elements can access their **private address spaces** and **exchange messages**



MIMD Hardware Taxonomy

MIMD

SM-MIMD	**DM-MIMD**
e.g. Multicore CPUs	*e.g. Clusters*

SM-MIMD

e.g. Multicore CPUs

- Low interaction overhead due to high coupling between processing elements

~ **Shared Memory Parallelism**

Covered in chapter B.

DM-MIMD

e.g. Clusters

- Highly scalable due to low coupling between processing elements

~ **Shared Nothing Parallelism**

Covered in chapter D.

Terminology		
shared memory system	vs.	distributed memory system
SM-MIMD	vs.	DM-MIMD
Multiprocessor	vs.	Multicomputer
see [Tanenbaum1985], [Foster1995], [Pfister1998]		

SM-MIMD Hardware

Processing elements can directly access a **common address space**

- **Uniform memory access (UMA) system**

  Processing elements observe the same memory access characteristics over the entire memory.

  - Simple to program against, but scalability issues

- **Non-uniform memory access (NUMA) system**

  Processing elements have different access characteristics for different memory regions

  - Scales well, but unaware programs can exhibit performance issues

SM-MIMD Hardware

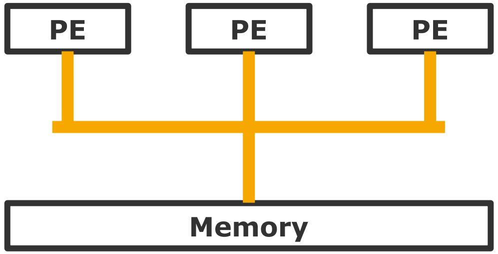
MIMD

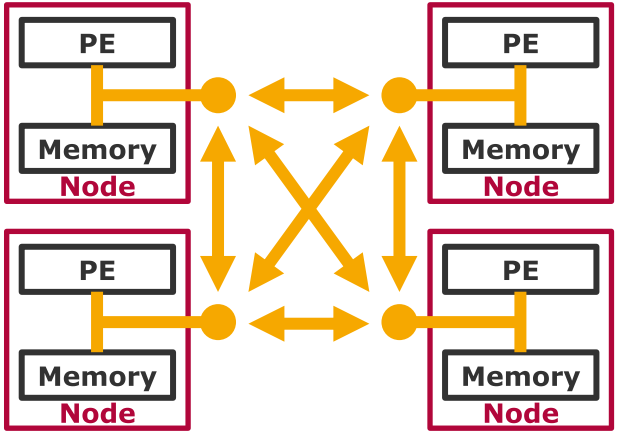
SM-MIMD
(Shared Memory)

DM-MIMD
(Distributed Memory)

UMA
(Uniform Memory Access)

NUMA
(Non-Uniform Memory Access)





DM-MIMD Hardware

Processing elements can access their **private address spaces** and **exchange messages**

Cluster: *Multiple independent machines connected through a network*

- **Compute** cluster: Speedup
- **Load Balancing** cluster: Throughput
- **High Availability** cluster: Dependability

All clusters are distributed systems, but only compute clusters intended for parallel workloads.

This lecture considers only compute clusters.

DM-MIMD Hardware

Simple way of scaling available compute resources:

Just connect multiple machines in a network.

Dominant architecture for High-End Systems:

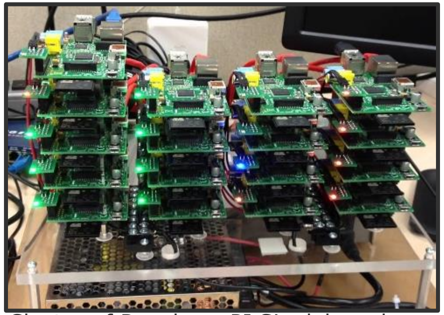
Especially High-Performance Computing

1995 *Toy Story* Render Farm
117 nodes × 2 CPUs = 234 CPUs

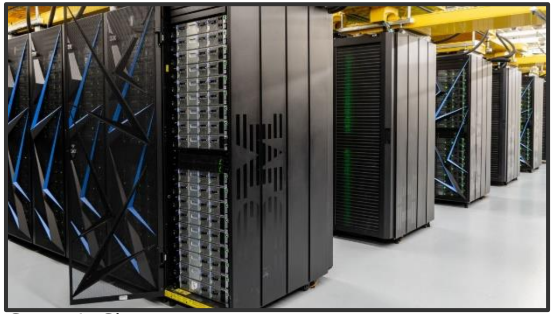
2001 *Monsters Inc.* Render Farm
250 nodes × 14 CPUs = 3500 CPUs

2019 Summit cluster (TOP500 #1 in 2019)
4608 nodes, 2 PB RAM, 10 MW power
× 2 CPUs × 22 Cores = 202 752 Cores
× 6 GPUs = 27 648 GPUs

Cluster of Desktop Computers



Cluster of RaspberryPI Singleboard Computers



Summit Cluster

Literature

[Flynn1966]

"Very High-Speed Computing Systems" Flynn, Michael J. Proceedings of the IEEE 54.12 (1966) IEEE

[Tanenbaum 1985]

"Distributed Operating Systems" Tanenbaum, Andrew S and Van Renesse, Robbert. ACM Computing Surveys 17.4 (1985) ACM

[Foster1995]

"Designing and Building Parallel Programs" Foster, Ian (1995) Addison-Wesley

[Pfister1998]

"In Search of Clusters" Pfister, Gregory F. 2nd edition (1998) Prentice-Hall Inc

And now for a break and
a bowl of Sencha.

*or beverage of your choice