COMPUTING SYSTEMS

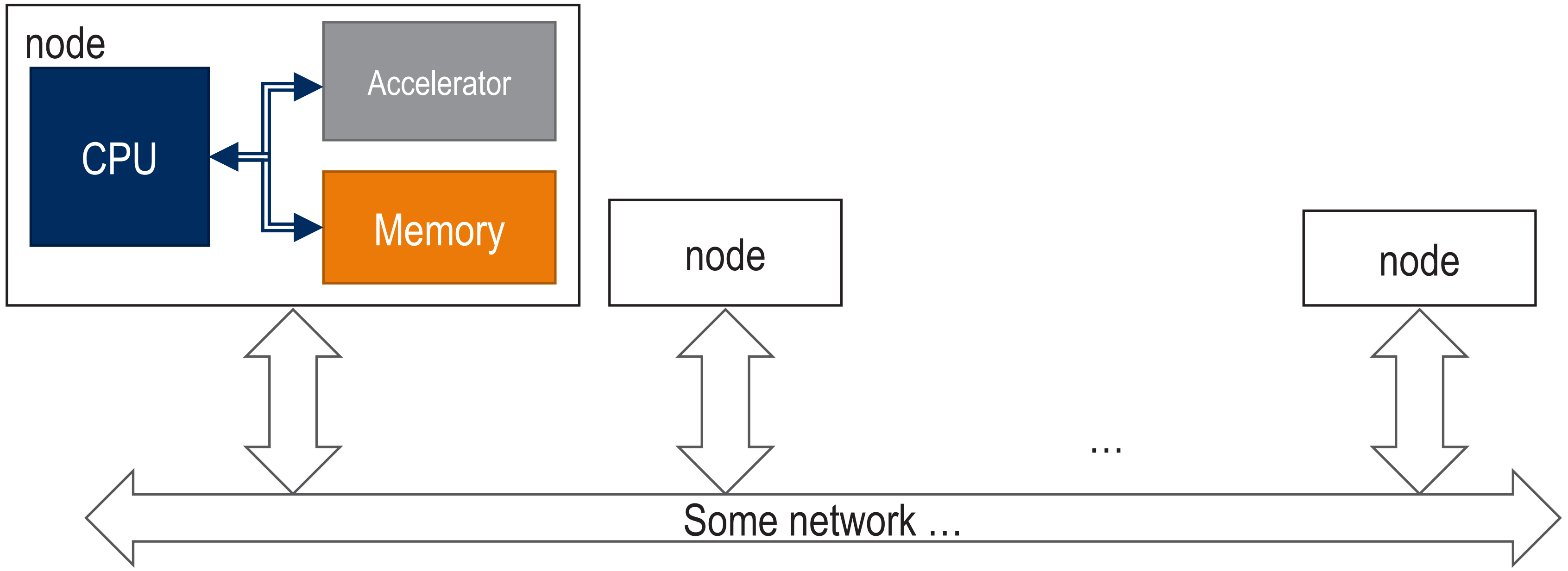
RECAP / INTRODUCTION

ANA-LUCIA VARBANESCU
CAES / EEMCS
a.l.varbanescu@utwente.nl

UNIVERSITY OF TWENTE.

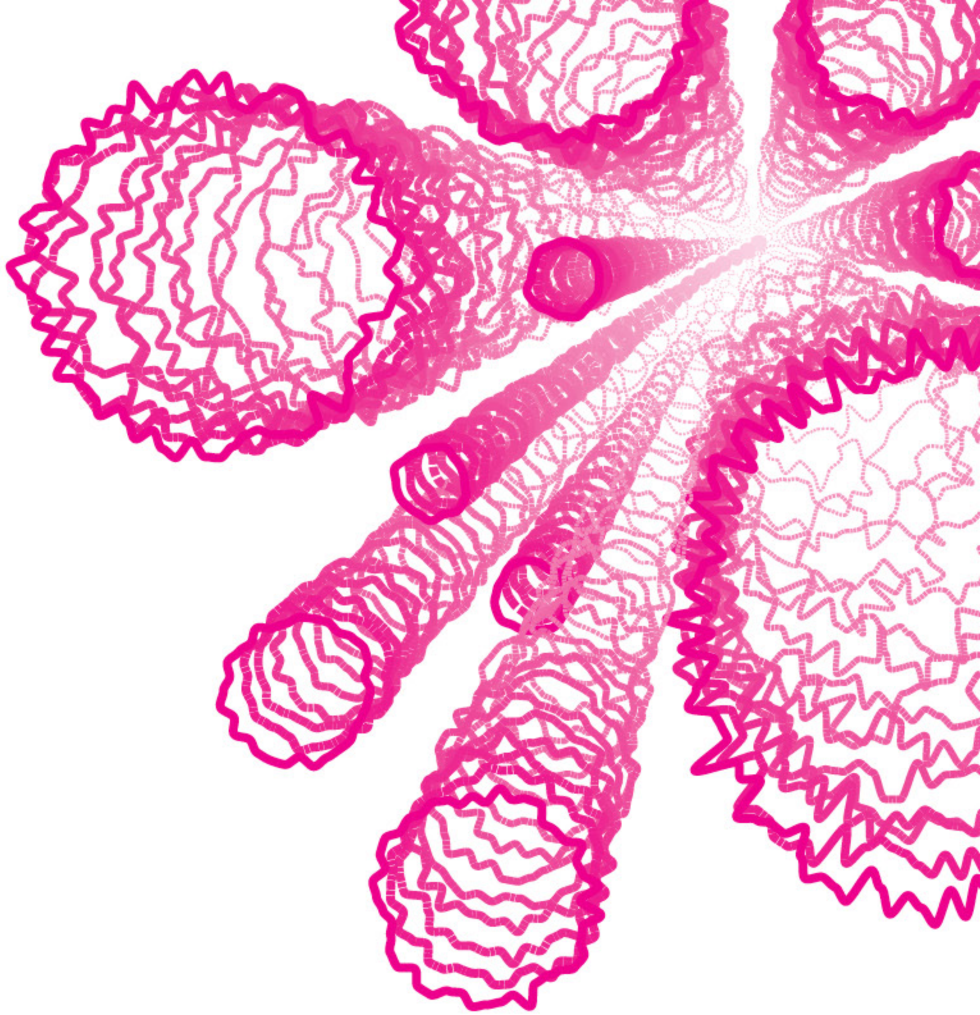
GENERIC VIEW

- Heterogeneous, parallel & distributed systems
- Performance / energy consumption / energy efficiency
- Single-node _and_ aggregate



UNIVERSITY OF TWENTE.

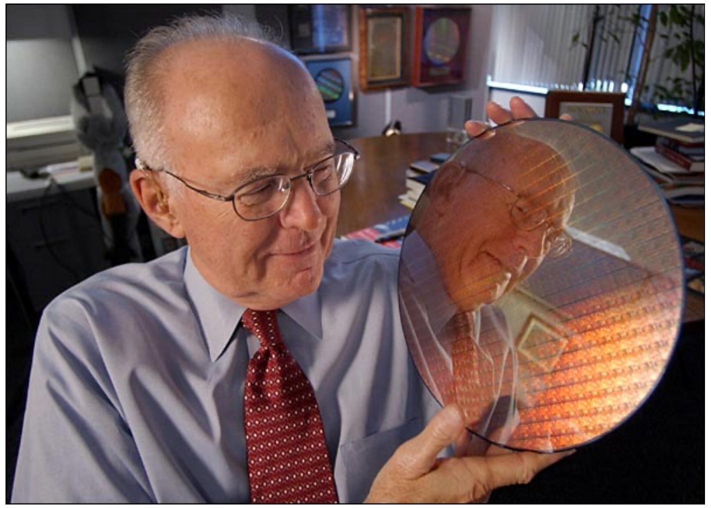
A ~~BIT~~ BYTE OF HISTORY

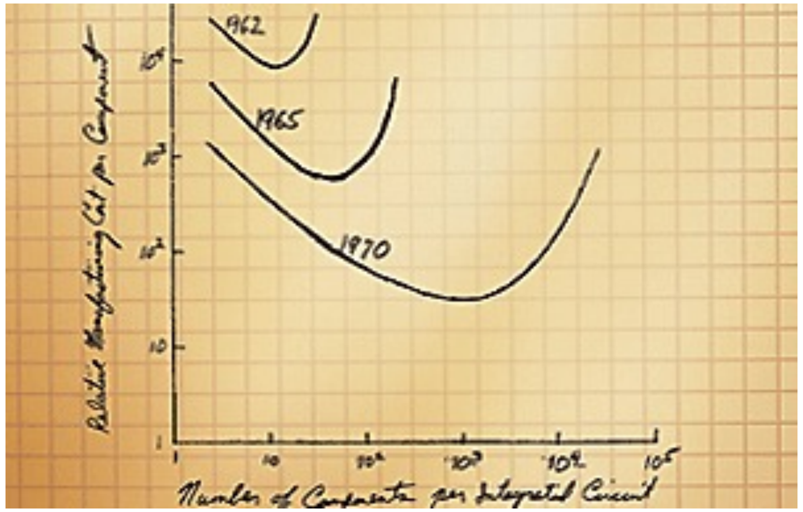


UNIVERSITY
OF TWENTE.

A TECHNOLOGY VIEW: MOORE'S LAW

- Gordon Moore (co-founder of Intel) predicted in 1965 that the transistor density of semiconductor chips would double roughly every 18 months.

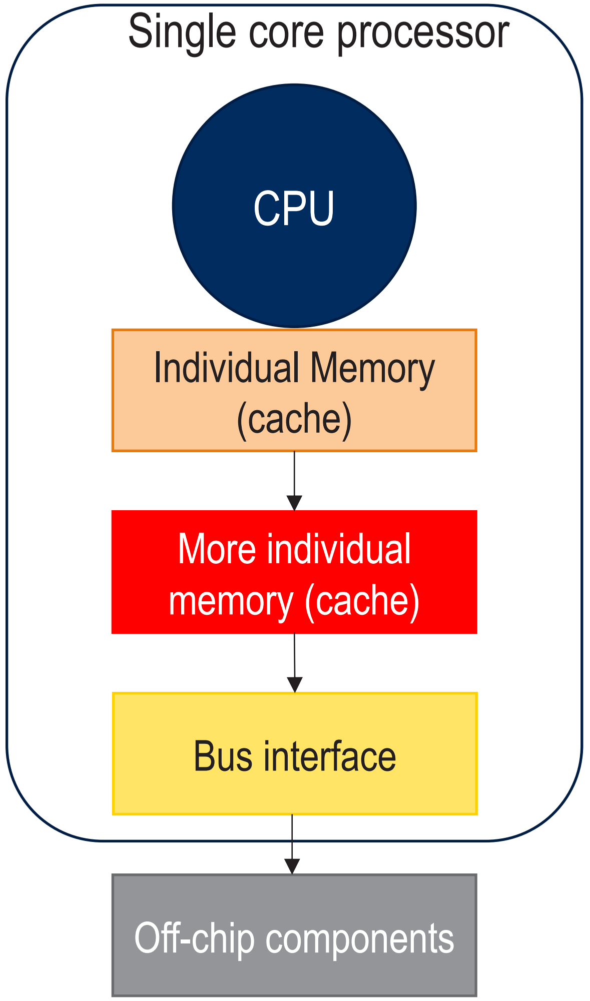



1962
1965
1970
10^4
10^3
10^2
10
1
10
10^2
10^3
10^4
10^5
Relative Manufacturing Cost per Component
Number of Components per Integrated Circuit

"The complexity for minimum component costs has increased at a rate of roughly a factor of two per year … Certainly over the short term this rate can be expected to continue, if not to increase…." Electronics Magazine 1965

TRADITIONALLY …

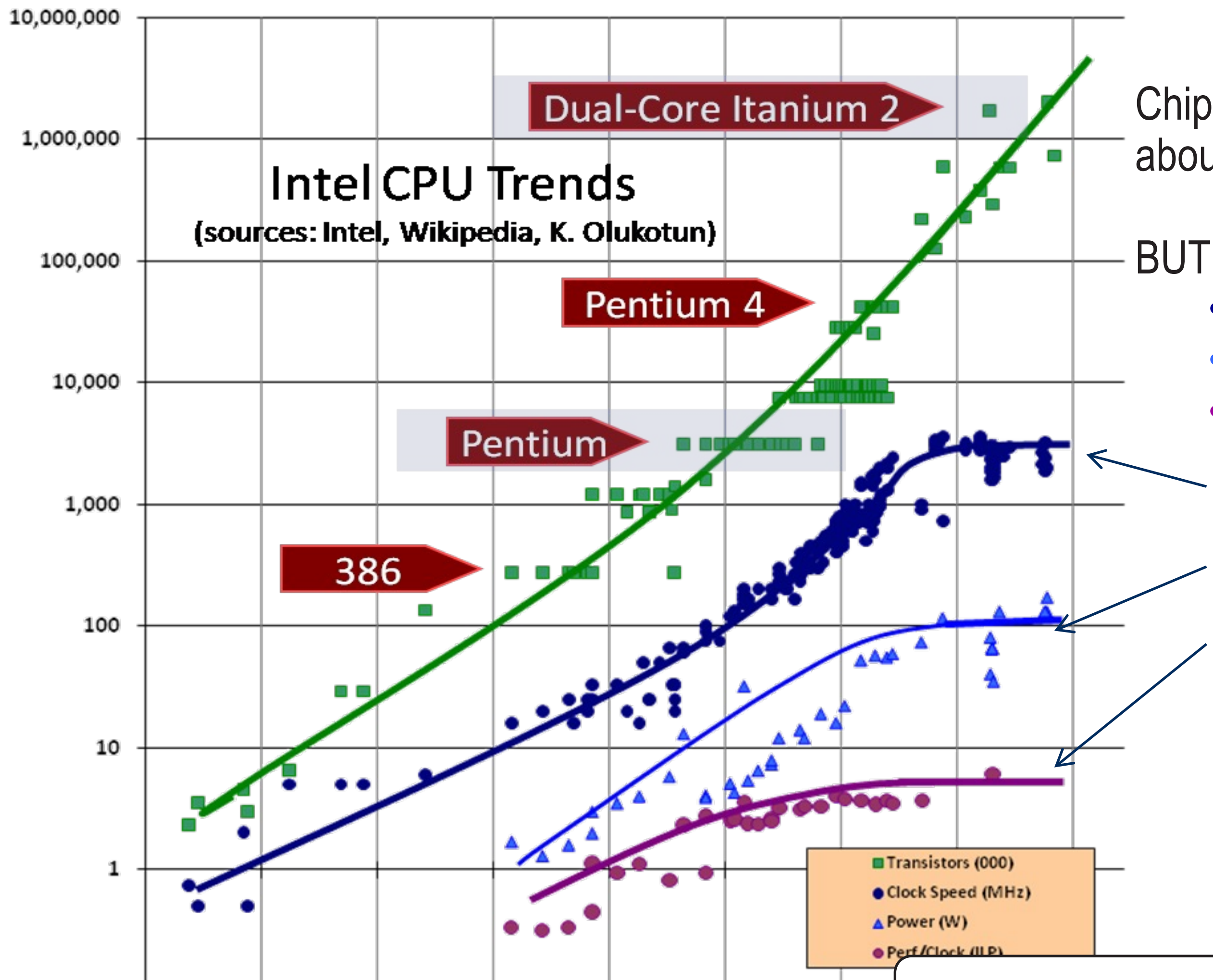
- More transistors = more functionality
- Improved technology = faster clocks = more speed
- Thus, every 18 months, we obtained better and faster processors.
- They were all sequential: they execute one operation per clock cycle.



Not anymore!
We no longer gain performance by "growing" sequential processors …

UNIVERSITY OF TWENTE.

EVOLUTION OF PROCESSORS



Chip density is continuing to increase about 2x every 2 years

BUT

- **Clock speed is not**
- **Power is not**
- **Instruction Level Parallelism is not**

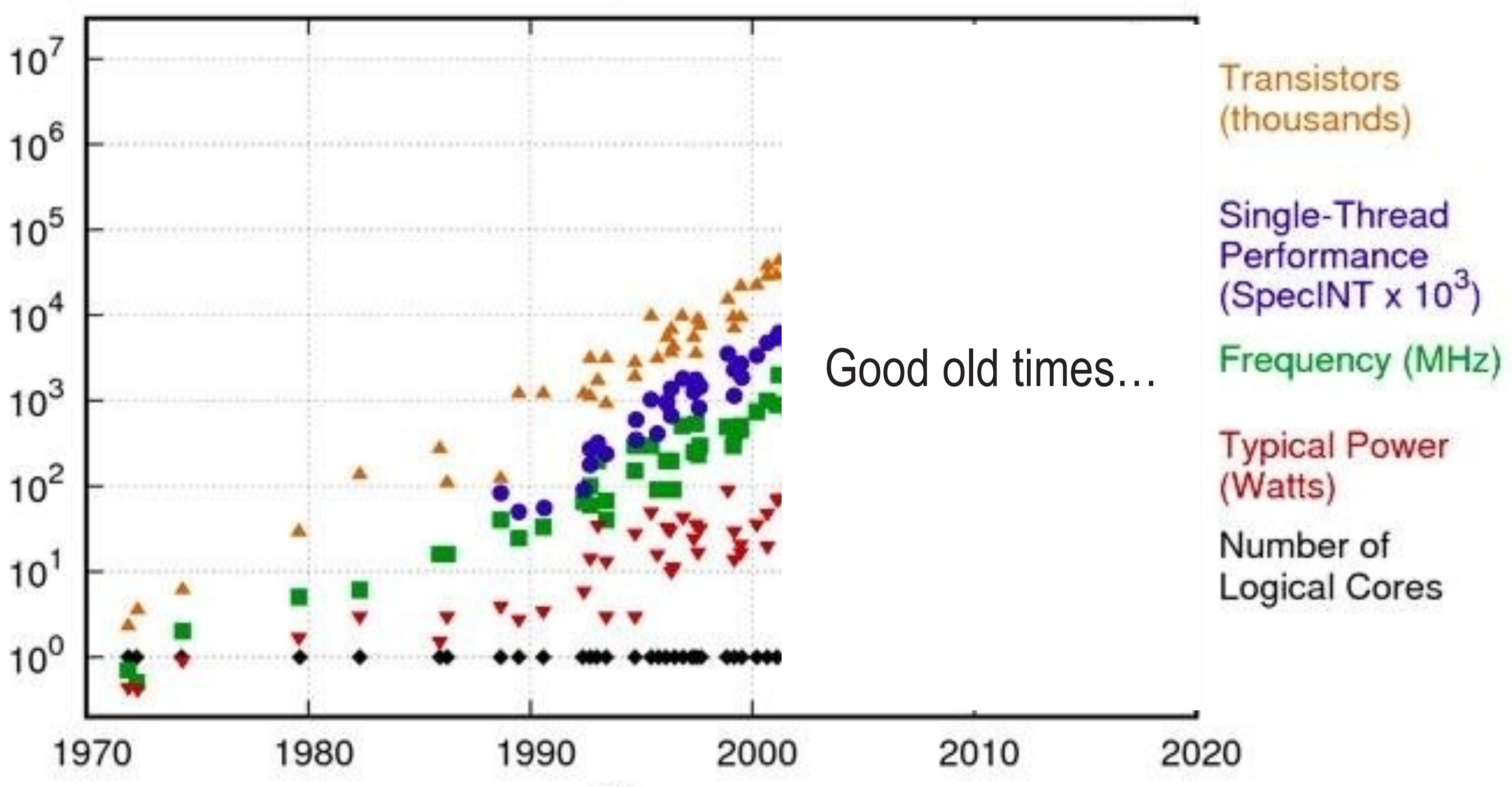
What does this mean in practice?

UNIVERSITY OF TWENTE.

NEW WAYS TO USE TRANSISTORS

Improve PERFORMANCE by using parallelism on-chip: multi-core (CPUs) and many-core processors (GPUs).

THE SHIFT TO MULTI-CORE

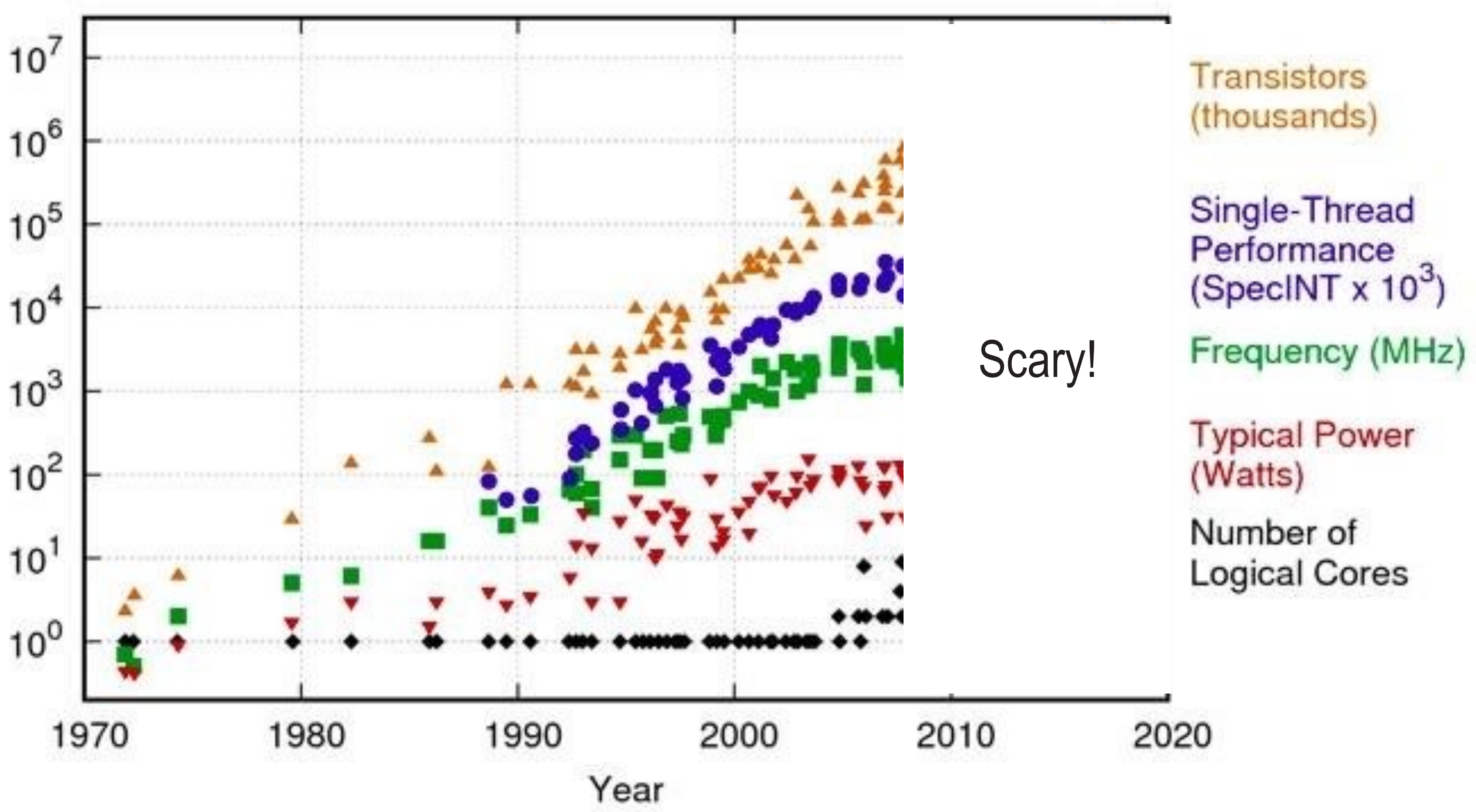


Good old times…

Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

UNIVERSITY OF TWENTE.

THE SHIFT TO MULTI-CORE

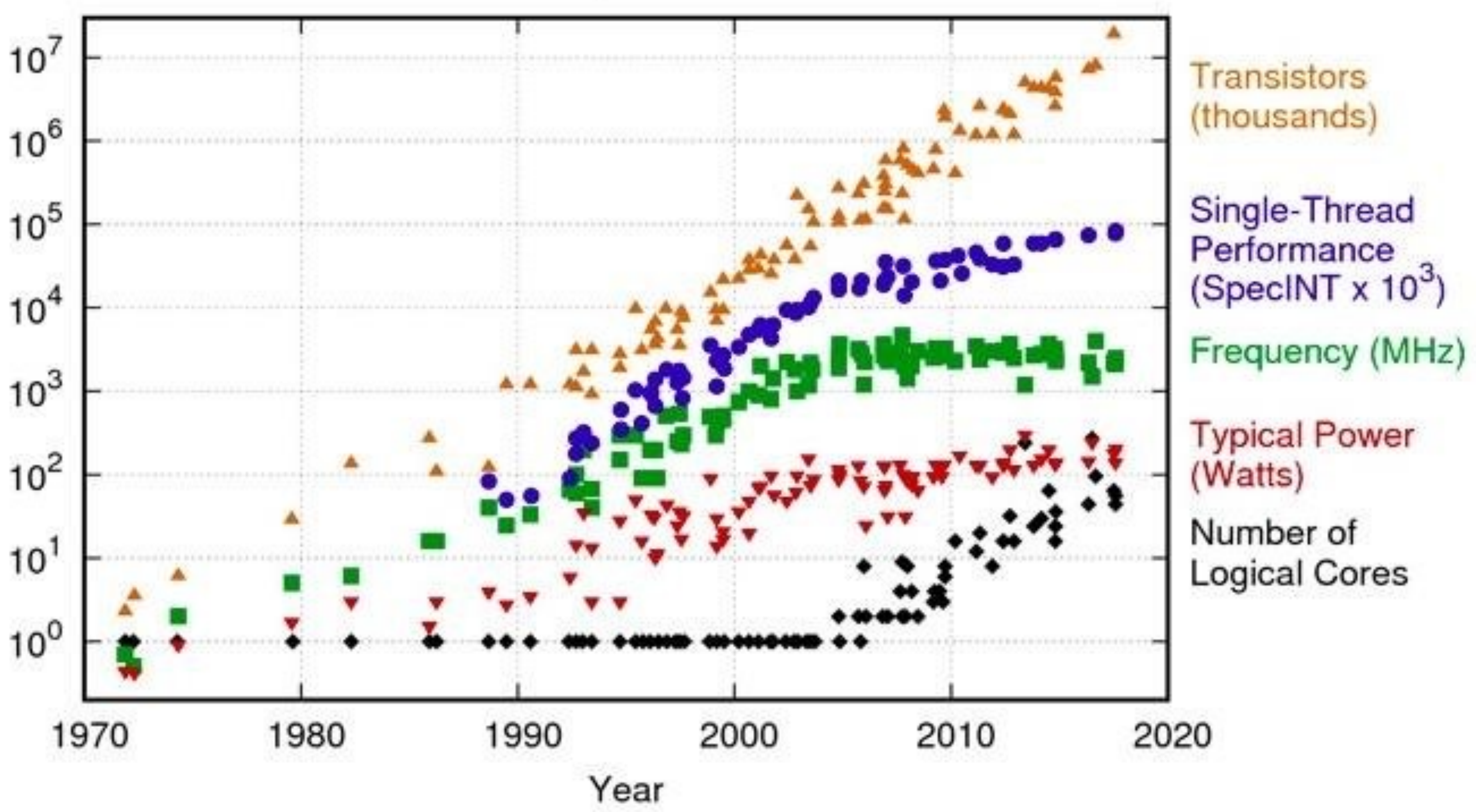


Scary!

Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

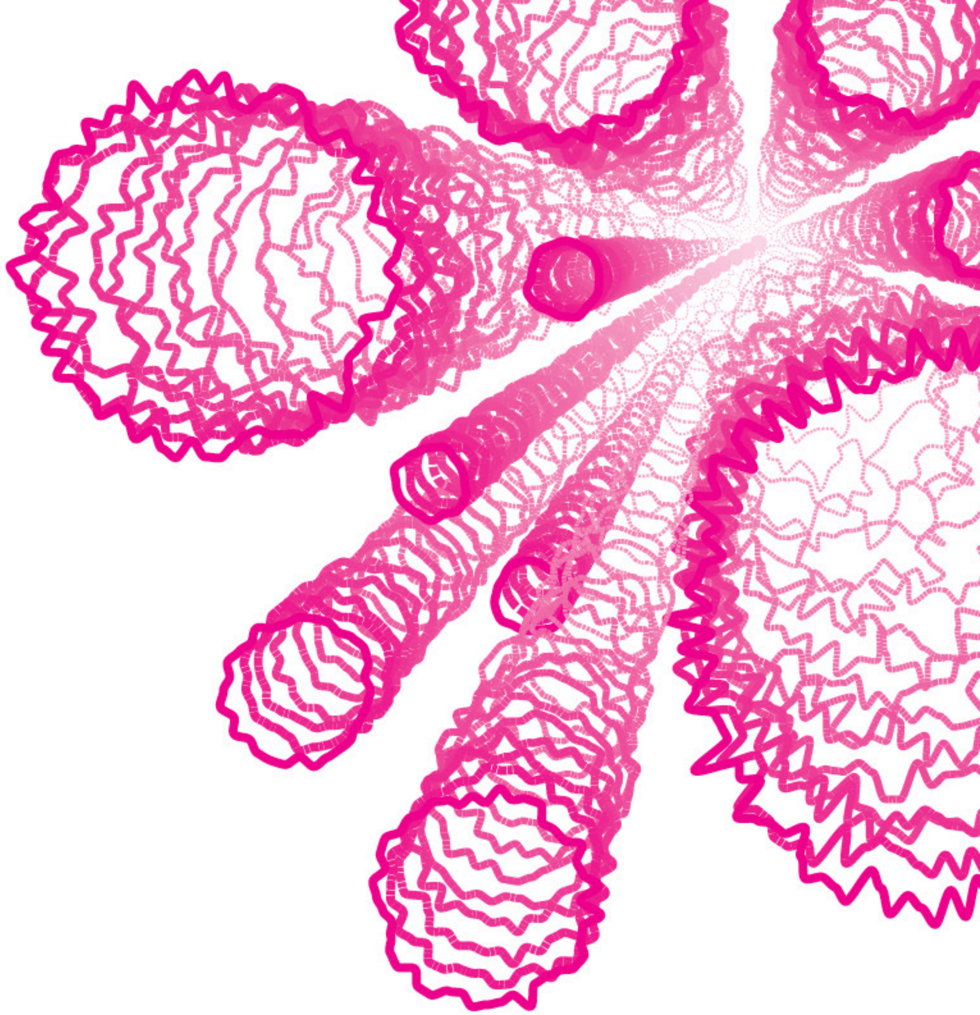
UNIVERSITY
OF TWENTE.

THE SHIFT TO MULTI-CORE



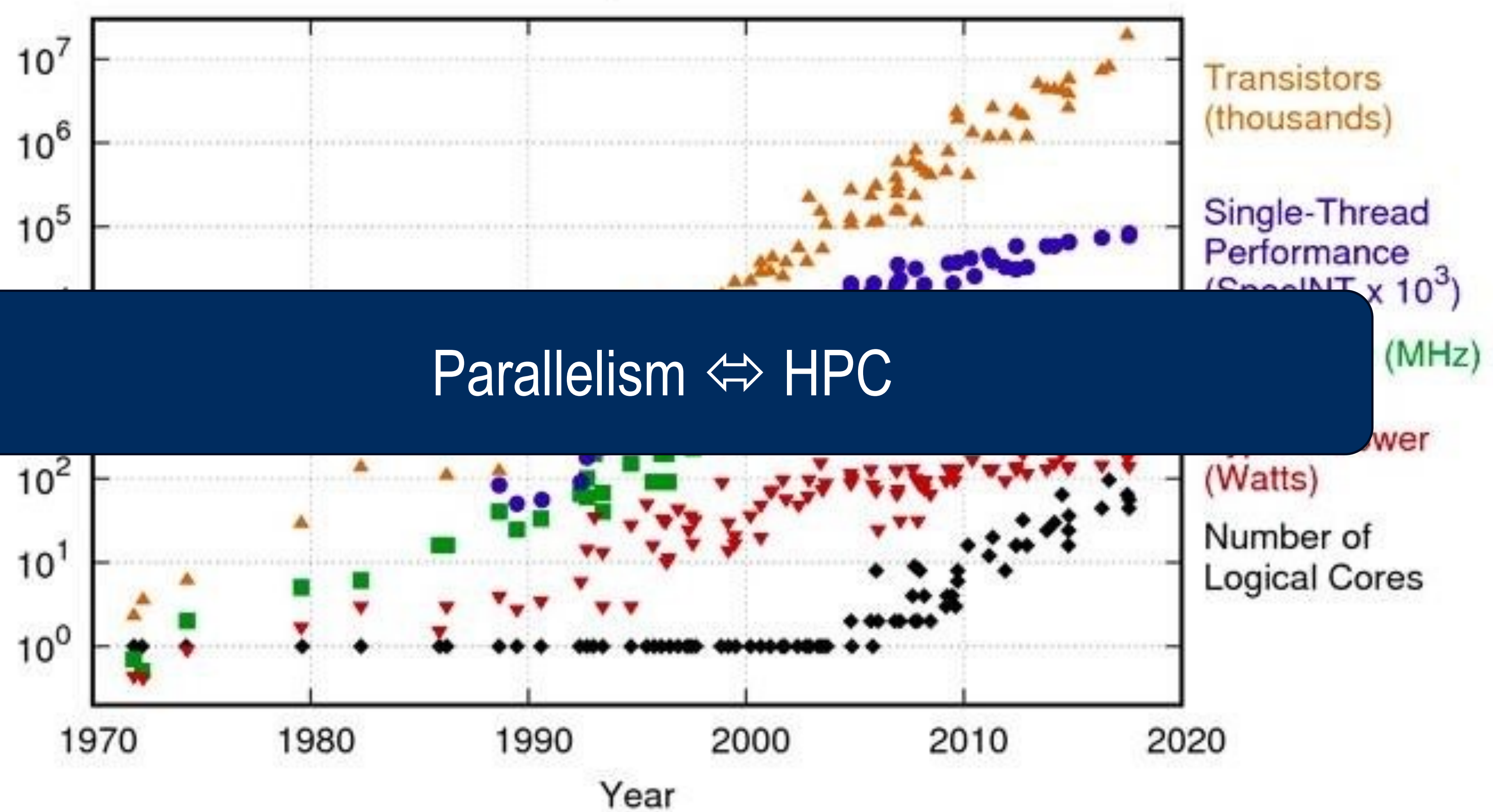
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

CURRENT PROCESSING UNITS



UNIVERSITY
OF TWENTE.

THE SHIFT TO MULTI-CORE

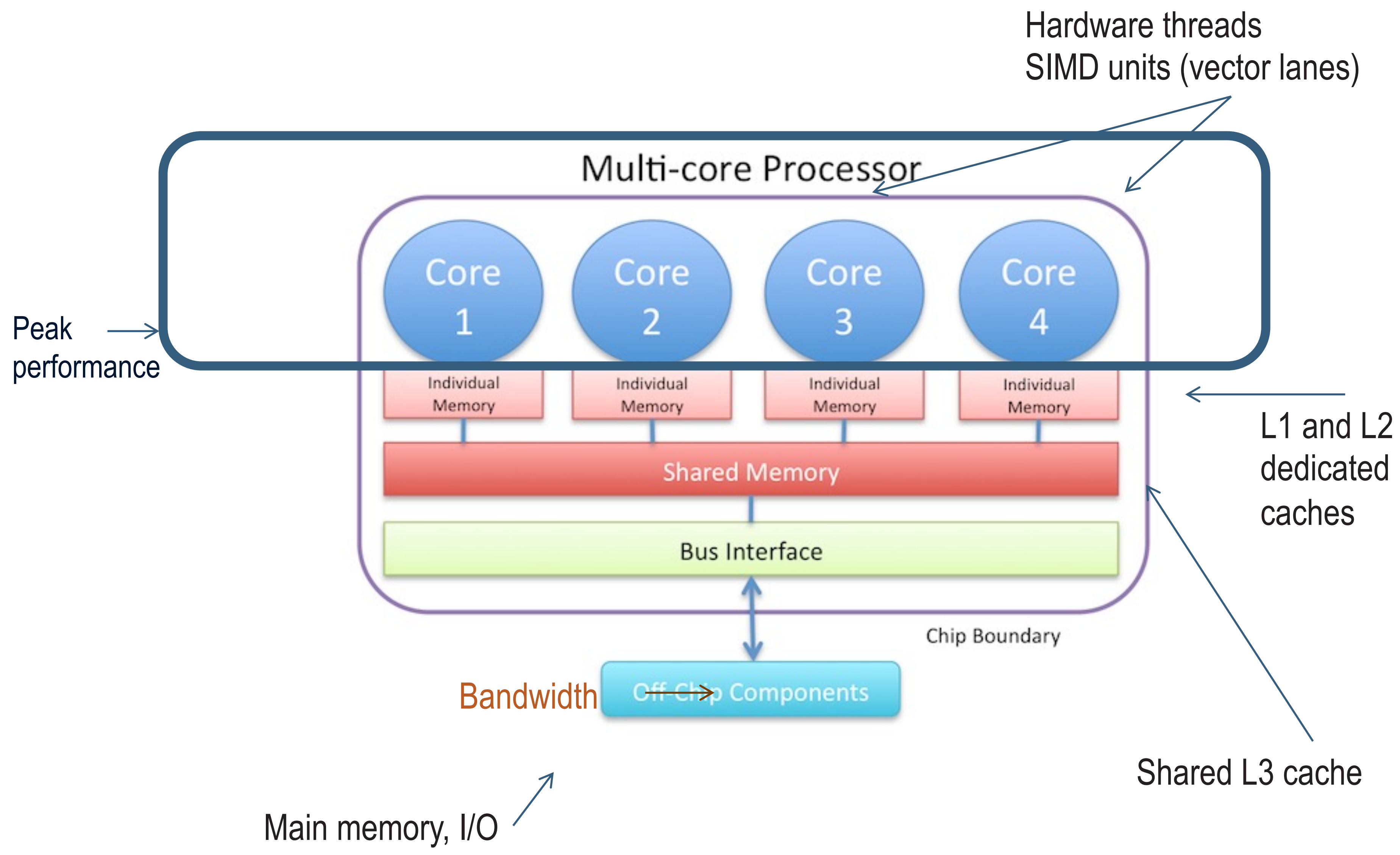


Parallelism ⇔ HPC

Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

UNIVERSITY OF TWENTE.

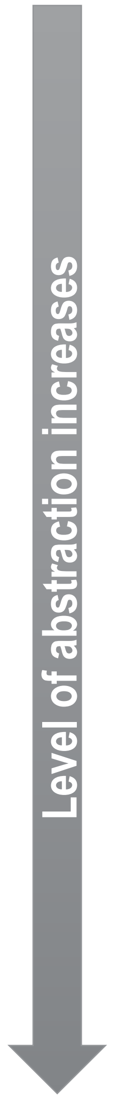
GENERIC MULTI-CORE CPU



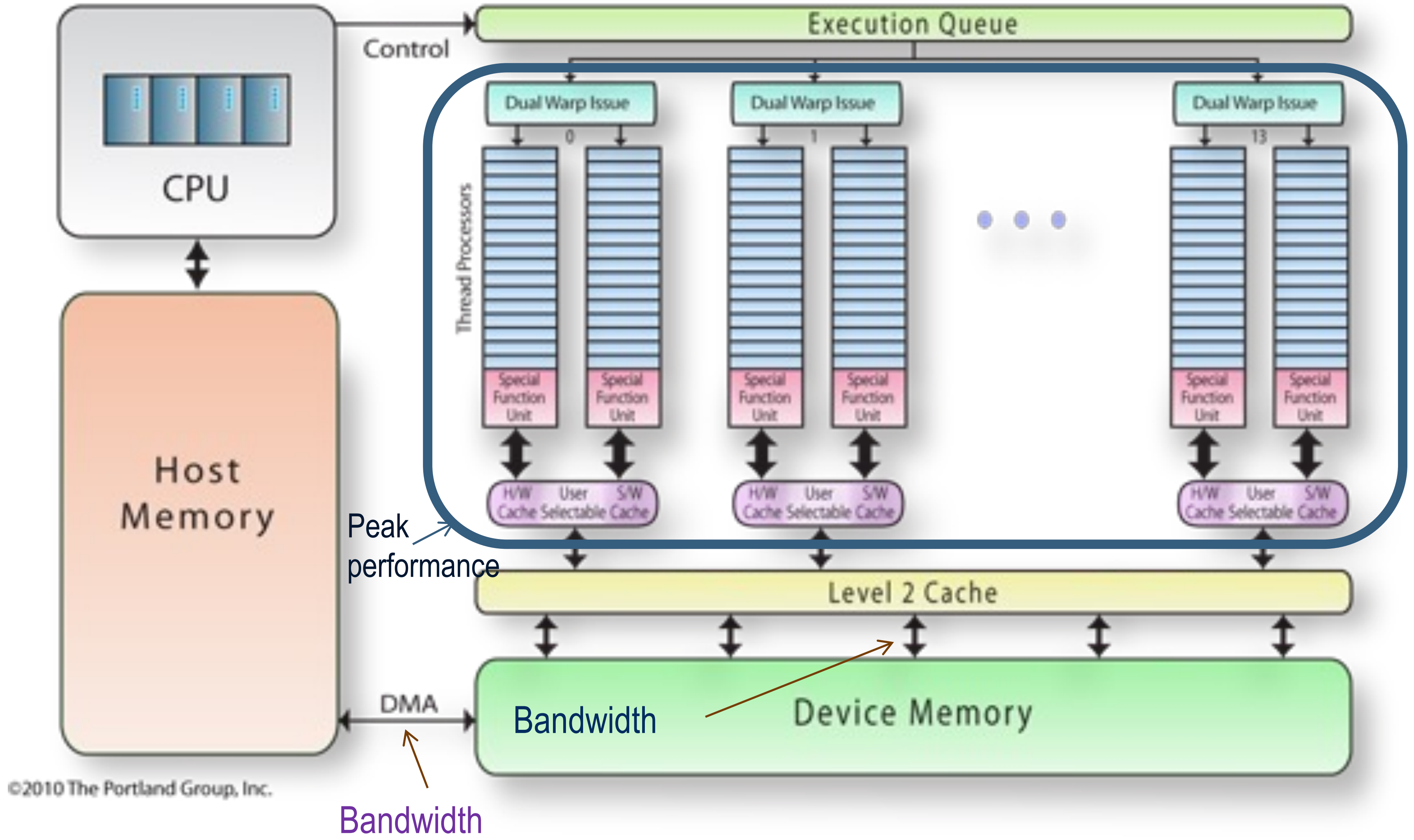
UNIVERSITY OF TWENTE.

(PARALLEL) PROGRAMMING MODELS

- Pthreads + intrinsics
- TBB – Thread building blocks
  - Threading library
- OpenCL
  - Can work, not great …
- SyCL
  - Update on OpenCL
- **OpenMP**
  - Traditional parallel library
  - High-level, pragma-based
- Cilk / Satin
  - Simple divide-and-conquer model
- OpenACC, OneAPI (from Intel)



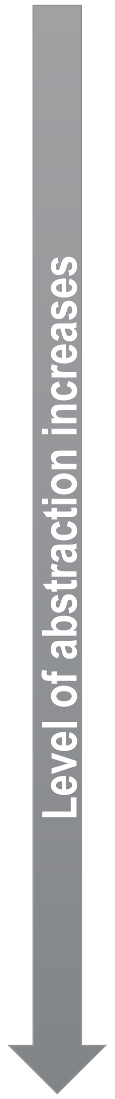
GENERIC GPU



UNIVERSITY OF TWENTE.

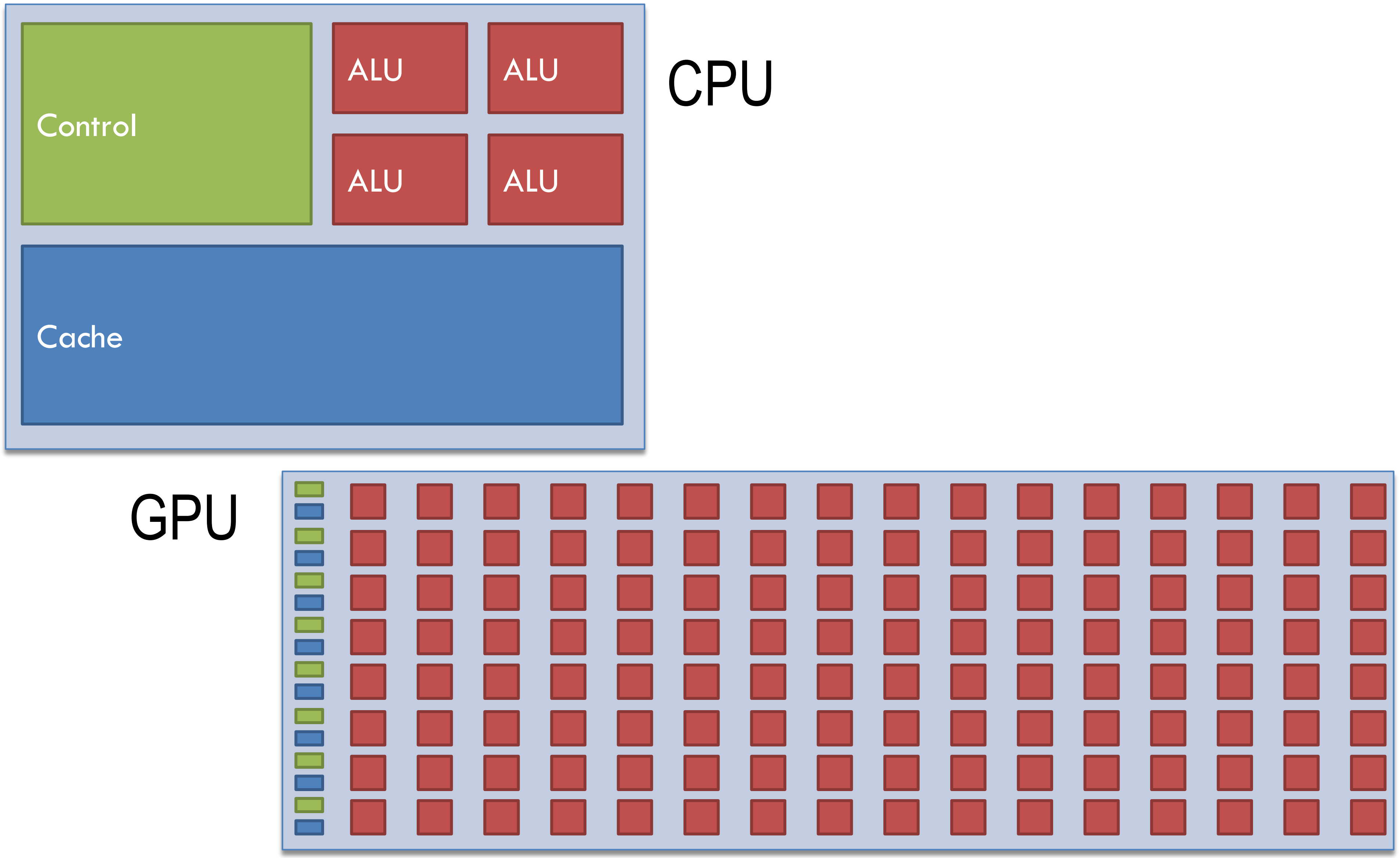
PROGRAMMING MODELS

- CUDA
  - NVIDIA proprietary
  - Easy for programming
  - Difficult for performance engineering
- OpenCL
  - Open standard
  - Portable
  - Difficult to write and debug
- SyCL
  - follow up of OpenCL : keep portability, add productivity
- OpenACC
  - OpenMP-like
  - High-level, pragma-based
- OpenMP
  - Starting from 4.5
  - Easy to use, cumbersome to understand
- Many other high-level programming models and tools



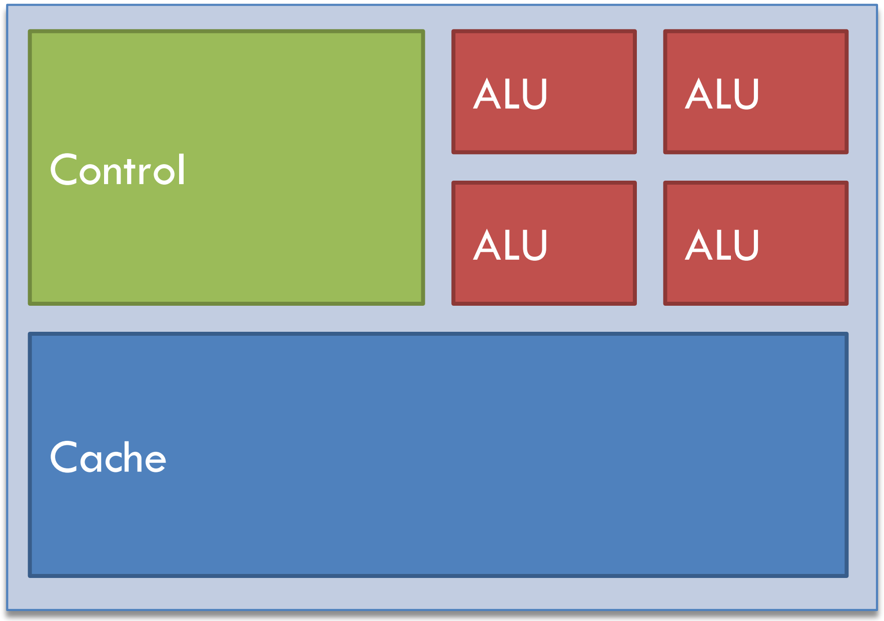
UNIVERSITY OF TWENTE.

CPU VS. GPU



UNIVERSITY OF TWENTE.

CPU VS. GPU

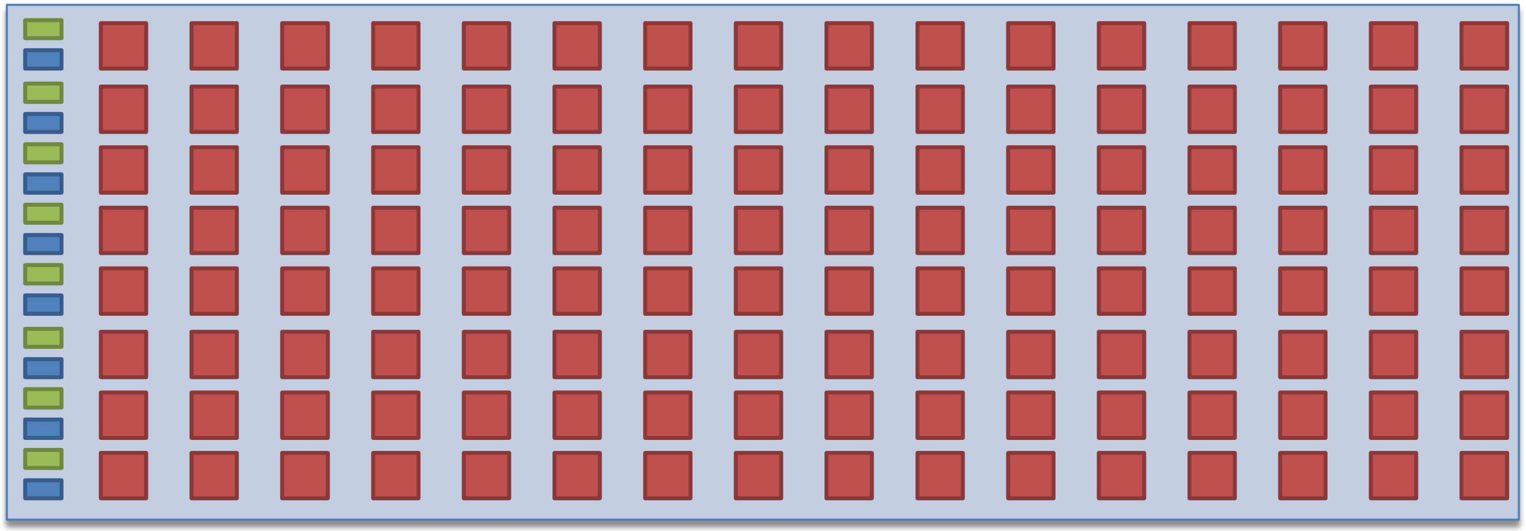


CPU

Low latency, high flexibility.
Excellent for irregular codes with limited parallelism.

GPU

High throughput.
Excellent for massively parallel workloads.



UNIVERSITY OF TWENTE.

PARALLELISM & CHALLENGES

- Parallel execution
    - Two or more applications/processes/tasks/jobs/instructions/… that can make progress in parallel.
- Challenges
    - Work and data
        - What tasks can run in parallel?
    - Ordering
        - Does the order of tasks matter?
    - Synchronization
        - Do the tasks need to wait for each other?

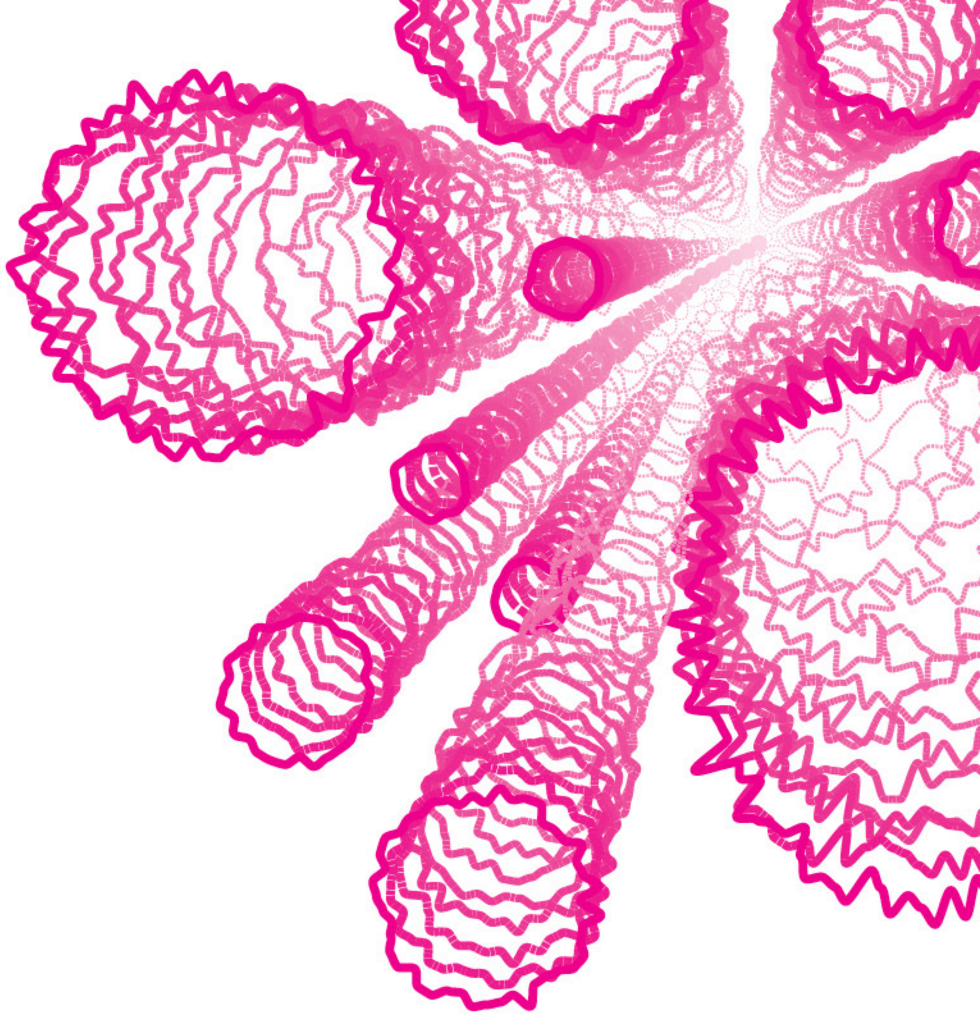
HPC ⇔ PARALLELISM

- HPC goal = maximize the performance of a given application on a given platform

- HPC = f(Machine, Application, Parallelism)
  - HPC machines
    - Super-chips, super-computers, grids, clouds
  - HPC applications
    - Lots of old and new application fields, simulations, predictions, models, ...
  - HPC parallelism
    - Multiple layers

UNIVERSITY
OF TWENTE.

PARALLEL SYSTEM MODELS

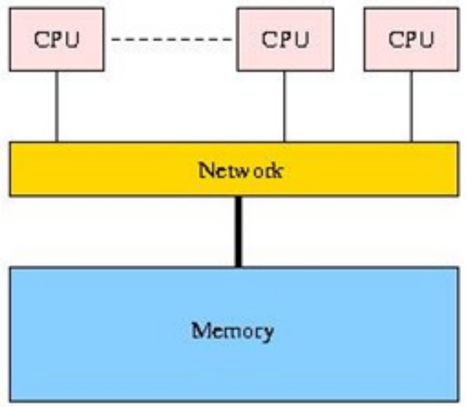


UNIVERSITY OF TWENTE.

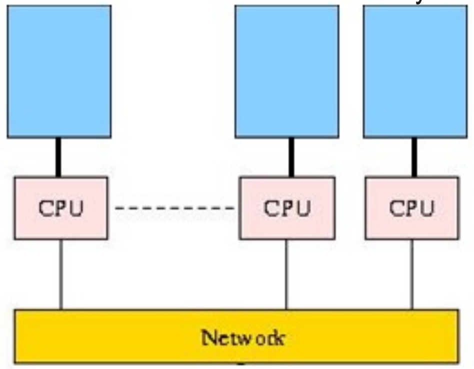
PARALLEL MACHINE MODELS

- Shared Memory
  - Multiple compute nodes
  - One single shared address space
  - Typical example: multi-cores
- Distributed Memory
  - Multiple compute nodes
  - Multiple, local (disjoint) address spaces
  - **Virtual shared memory:** software/hardware layer “emulates” shared memory
  - Typical example: clusters
- Hybrids
  - Multiple compute nodes, typically heterogeneous
  - Mixed address space(s), some shared, some global memory

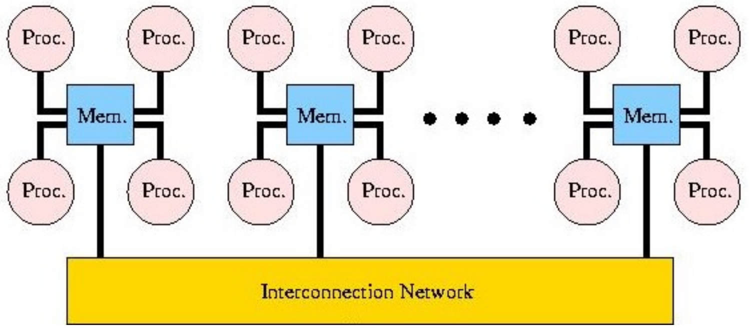
Shared memory



Distributed memory



Hybrid



PARALLEL MACHINE MODELS

- Shared Memory

Programming: multi-threading
Programming models: OpenMP, pthreads, TBB, …

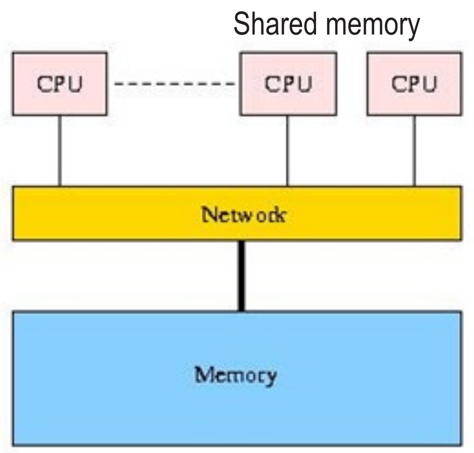
- Distributed Memory

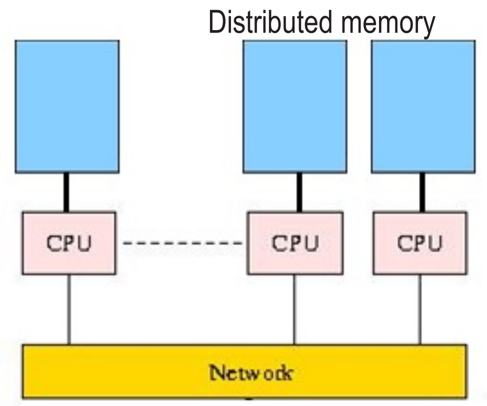
Programming: message passing
Programming models: MPI, Big-data models, …

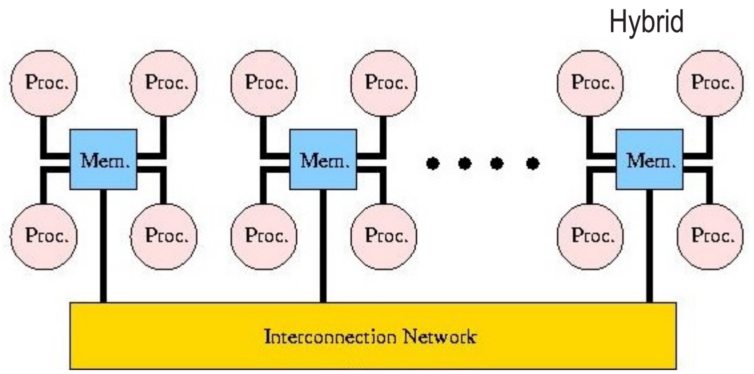
shared memory

- Hybrids

Programming: very diverse, depending on the hardware configuration







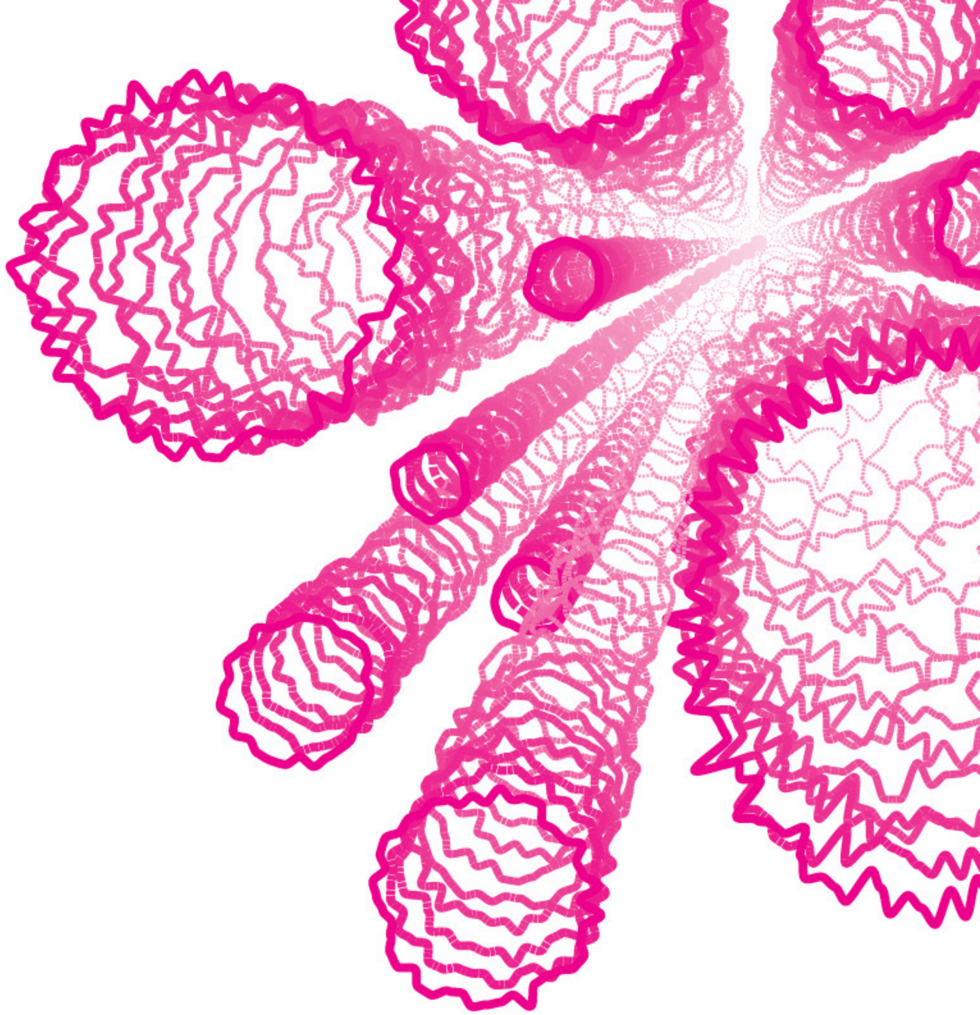
EXAMPLES

- Multi-core CPUs ?
  - Shared memory with respect to system memory
  - Hybrid when taking caches into account
- Clusters ?
  - Distributed memory
  - Could be shared if middleware for virtual shared space is provided
- Supercomputers ?
  - Usually hybrid
- GPUs ?
- Architectures with GPUs?
  - Distributed for traditional, off-chip GPUs
  - Shared for new APUs

MAJOR ISSUES

- Shared Memory model
  - Scalability problems (interconnect)
  - Programming challenge: RD/WR Conflicts
- Distributed Memory model
  - Data distribution is mandatory
  - Programming challenge: remote accesses, consistency
- Virtual Shared Memory model
  - Significant virtualization overhead
  - Easier programming
- Hybrid models
  - Local/remote data more difficult to trace

ZOOM-IN:
CORE-LEVEL FEATURES

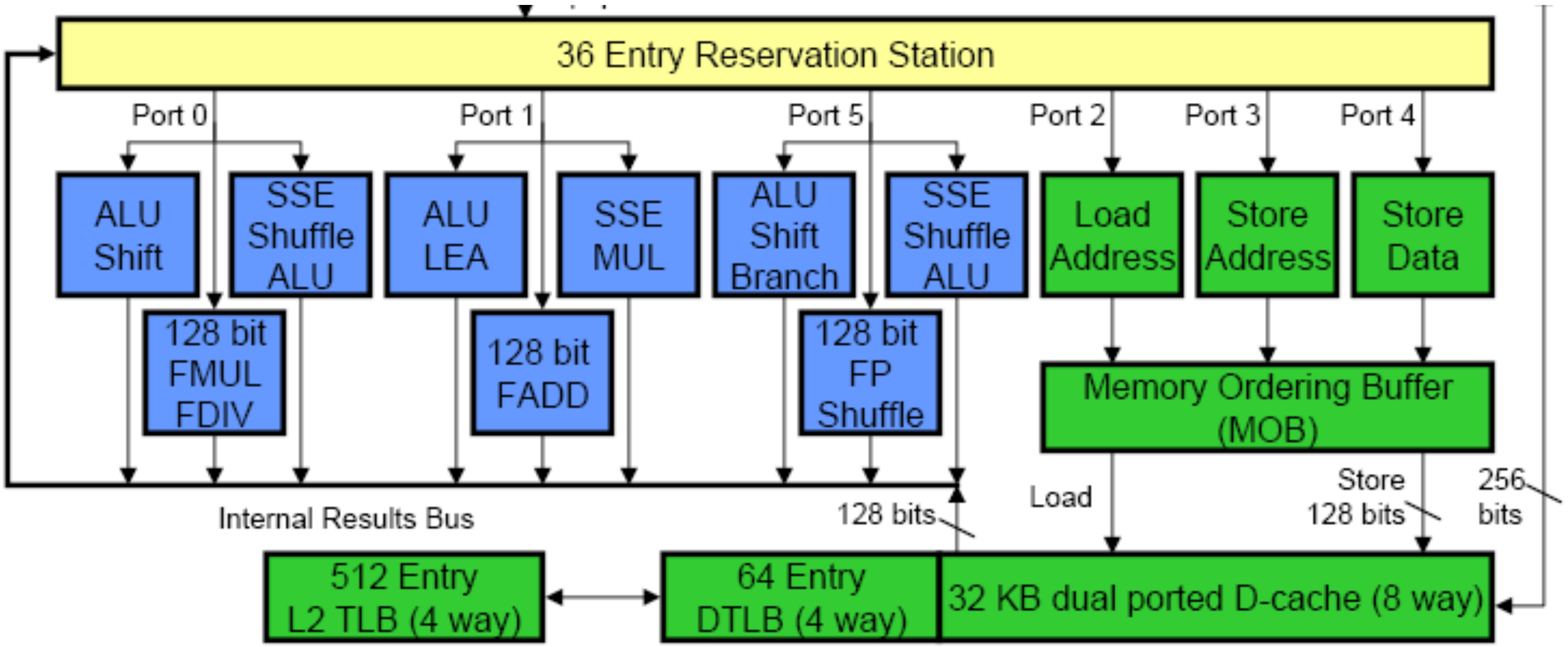


UNIVERSITY
OF TWENTE.

CORE-LEVEL CHALLENGES

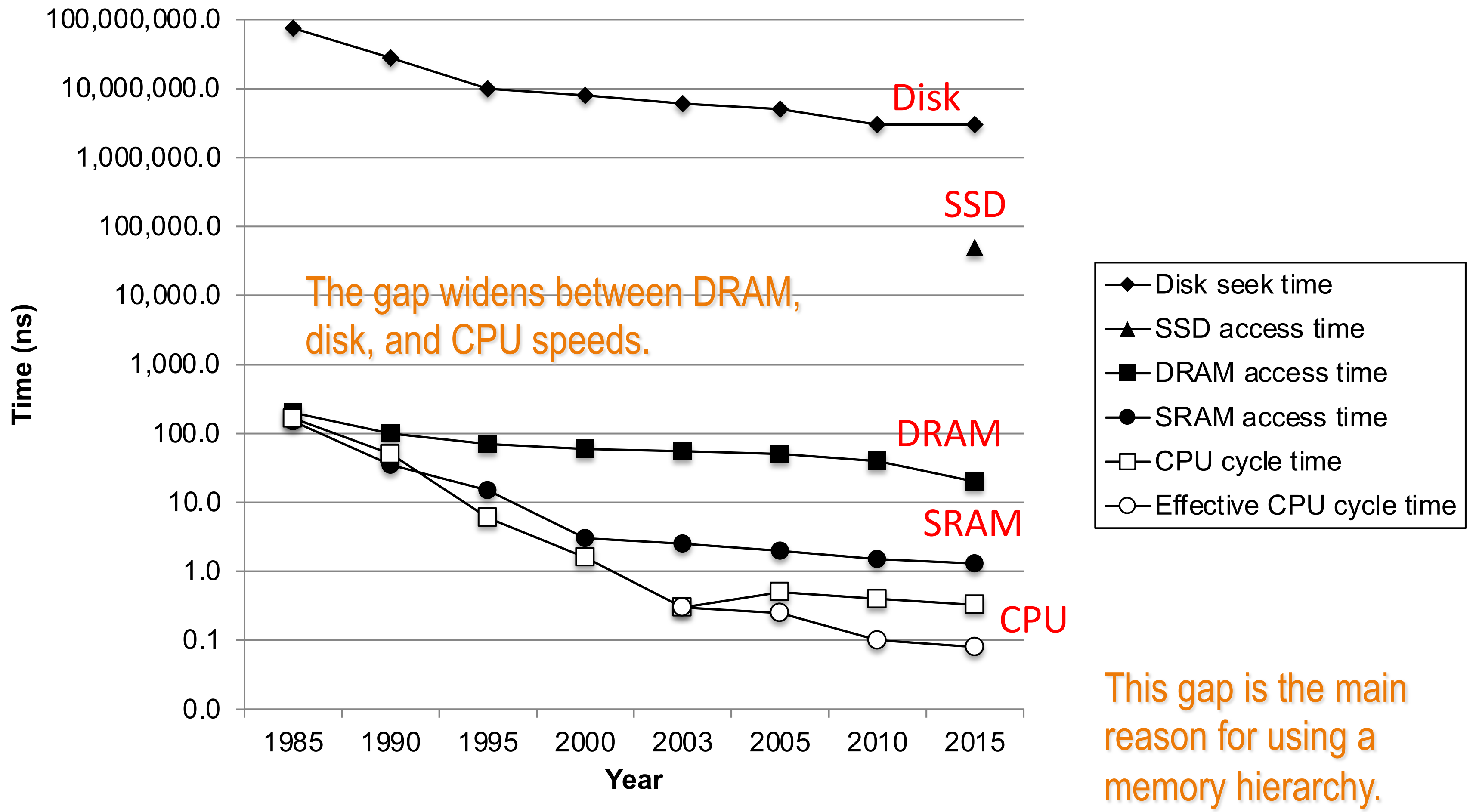
- Exploit low-level parallelism
  - ILP = instruction-level parallelism
  - SIMD = single-instruction multiple data
  - Hidden from programmers
    - Should be automatically addressed by compilers
    - Low-level languages do expose it, if needed
- Exploit memory hierarchies
  - Caches and non-caches alike

A REAL CPU …



UNIVERSITY OF TWENTE.

THE *PU-MEMORY GAP



The gap widens between DRAM, disk, and CPU speeds.

This gap is the main reason for using a memory hierarchy.

CACHES

- **Cache:** A smaller, faster storage device that acts as a staging area for a subset of the data in a larger, slower device.

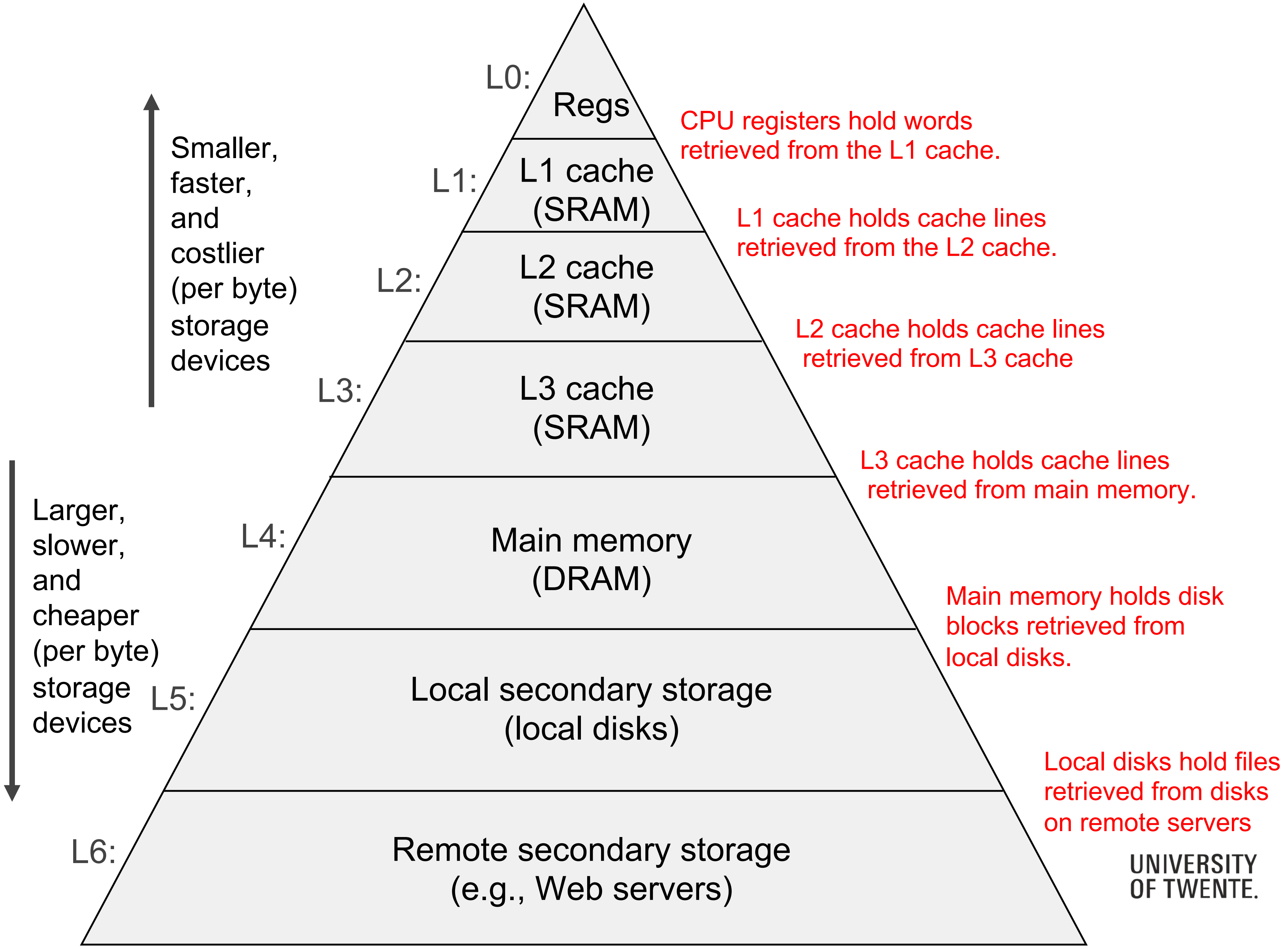
- Memory hierarchy
  - Multiple layers of memory, from **small & fast** (lower levels) to **large & slow** (higher levels)
  - *For each k, the faster, smaller device at level k is a cache for the larger, slower device at level k+1.*

- How/why do memory hierarchies work?
  - **Locality** => data at level k is used more often than data at level k+1.
    - Level k+1 can be slower, and thus larger and cheaper.

MEMORY HIERARCHY

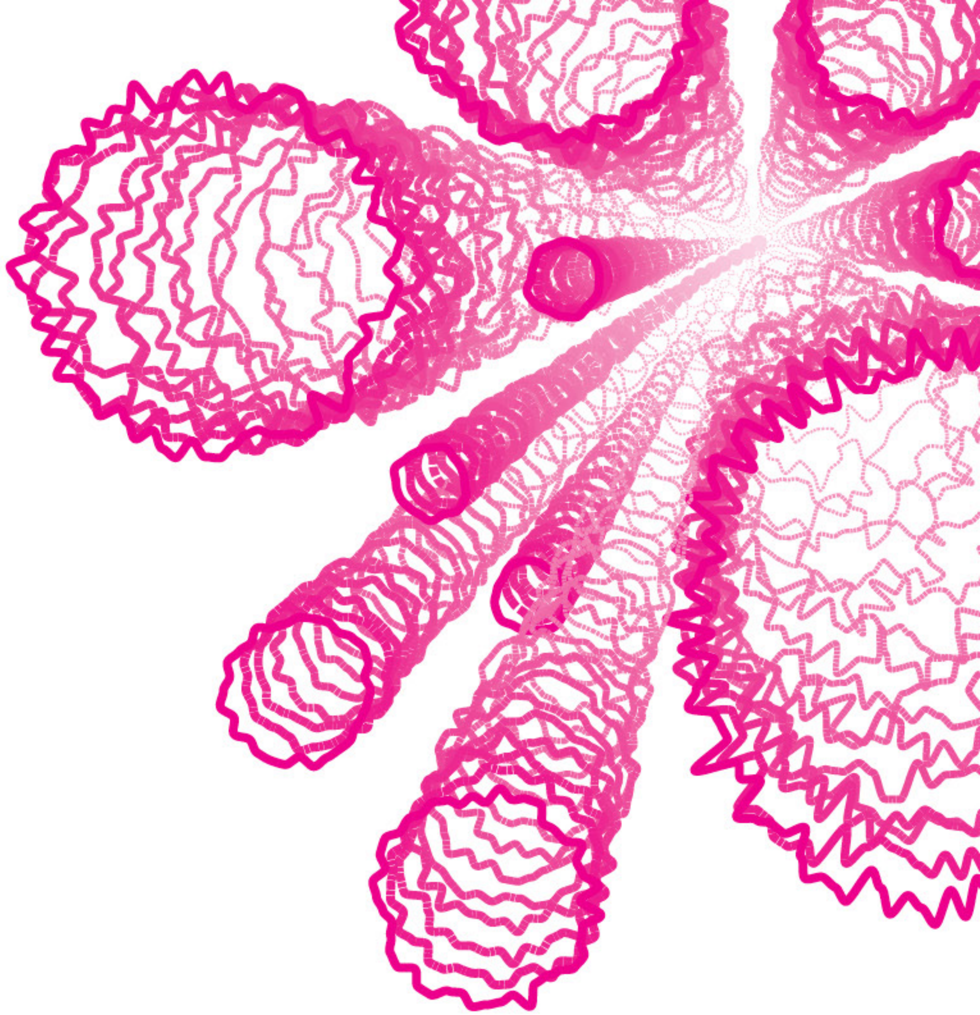
… works because of

- **Data reuse &**
- **Principle of locality**



Smaller, faster, and costlier (per byte) storage devices
Larger, slower, and cheaper (per byte) storage devices
L0:
Regs
L1:
L1 cache (SRAM)
L2:
L2 cache (SRAM)
L3:
L3 cache (SRAM)
L4:
Main memory (DRAM)
L5:
Local secondary storage (local disks)
L6:
Remote secondary storage (e.g., Web servers)
CPU registers hold words retrieved from the L1 cache.
L1 cache holds cache lines retrieved from the L2 cache.
L2 cache holds cache lines retrieved from L3 cache
L3 cache holds cache lines retrieved from main memory.
Main memory holds disk blocks retrieved from local disks.
Local disks hold files retrieved from disks on remote servers

ZOOM-OUT:
MULTI-NODE SYSTEMS



UNIVERSITY
OF TWENTE.

PUTTING IT ALL TOGETHER

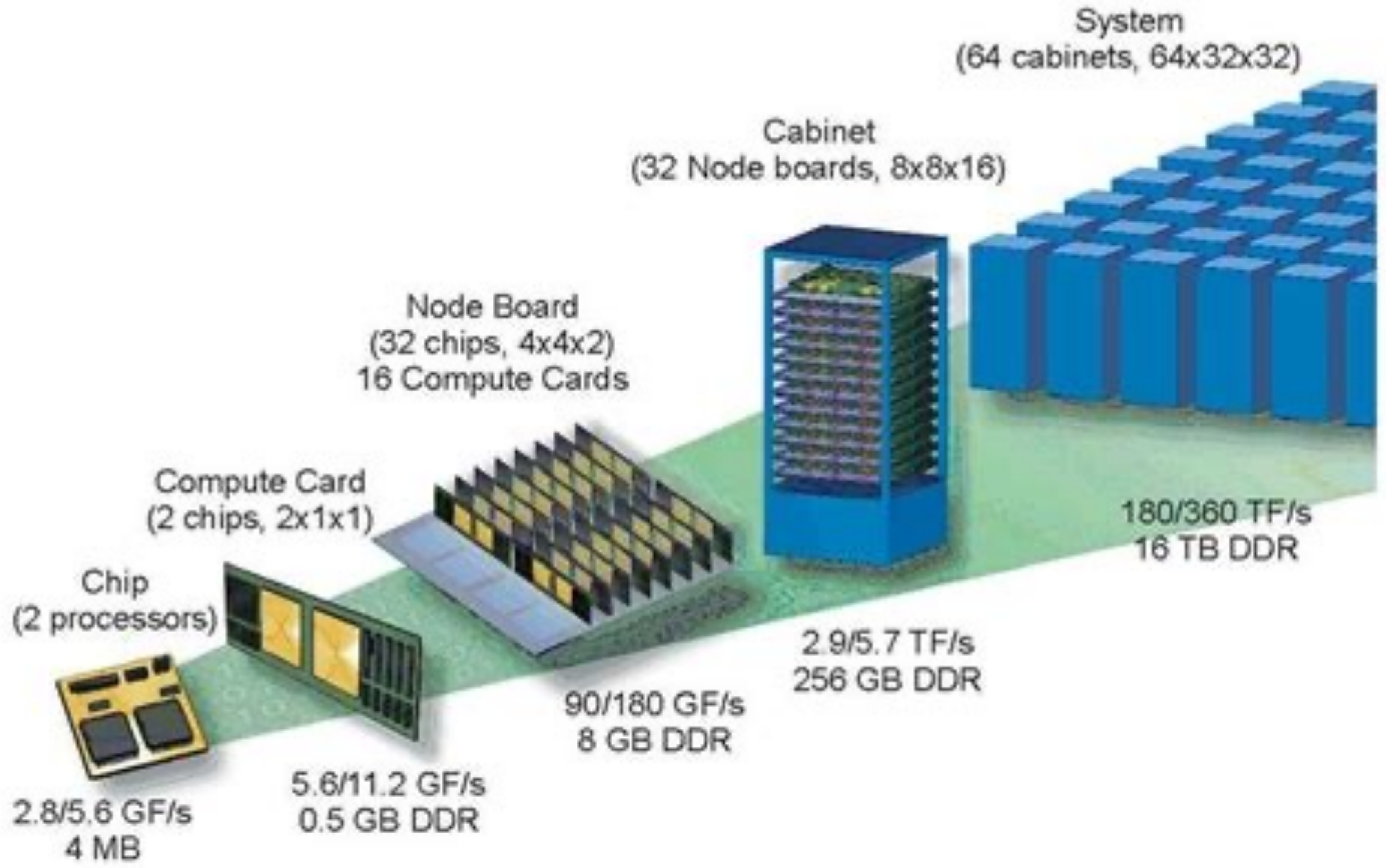
- Multiple nodes
    - Potentially grouped/clustered in islands
- Communication network
    - Latency & throughput differences compared to intra-node
- Homogeneous vs. Heterogeneous

- Peak Perfomance: summing all up
- Energy consumption: summing it all up

- Measurements: summing it all up .. sort of ☺

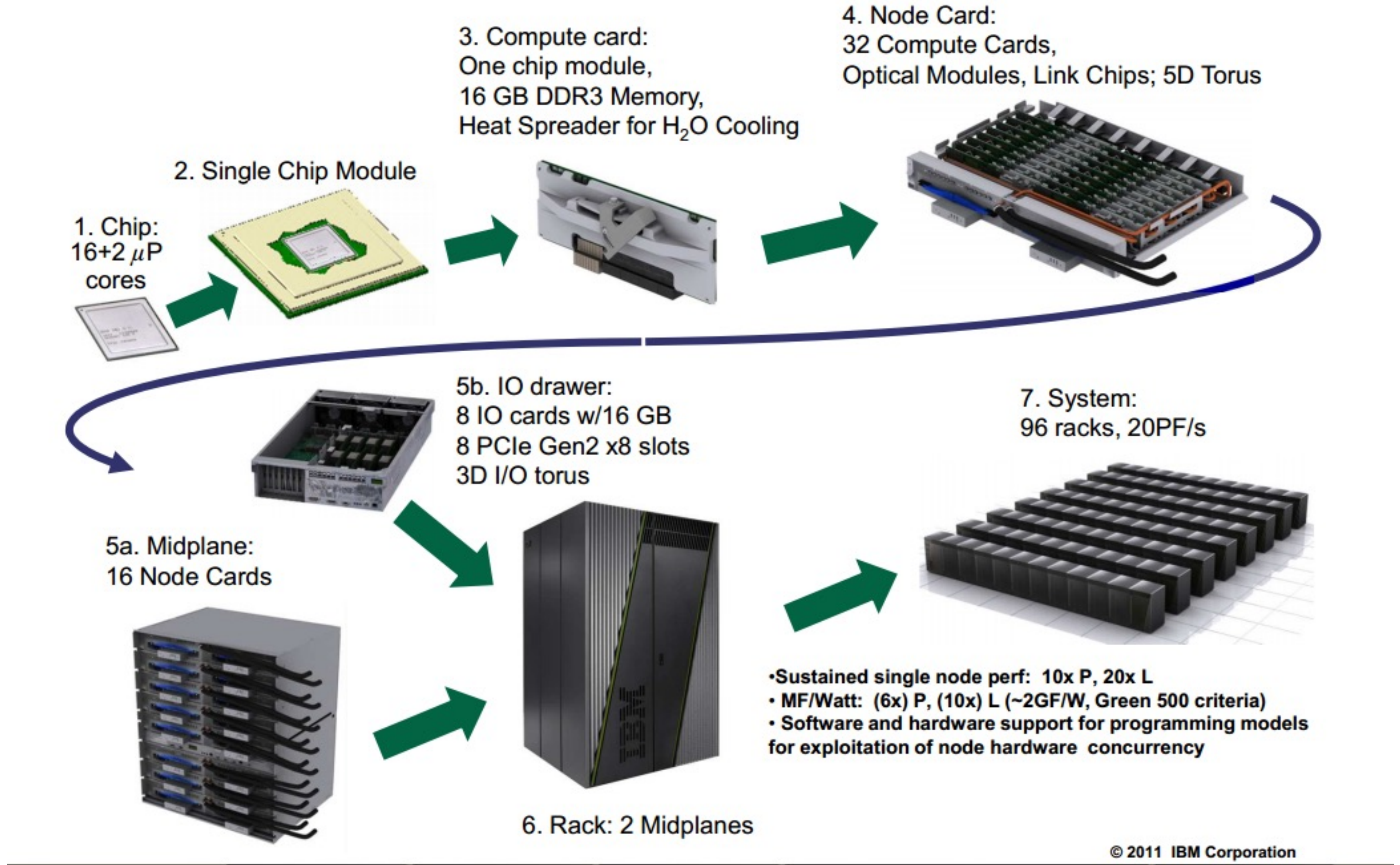
UNIVERSITY
OF TWENTE.

PUTTING IT ALL TOGETHER: IBM’S BLUGENE/L



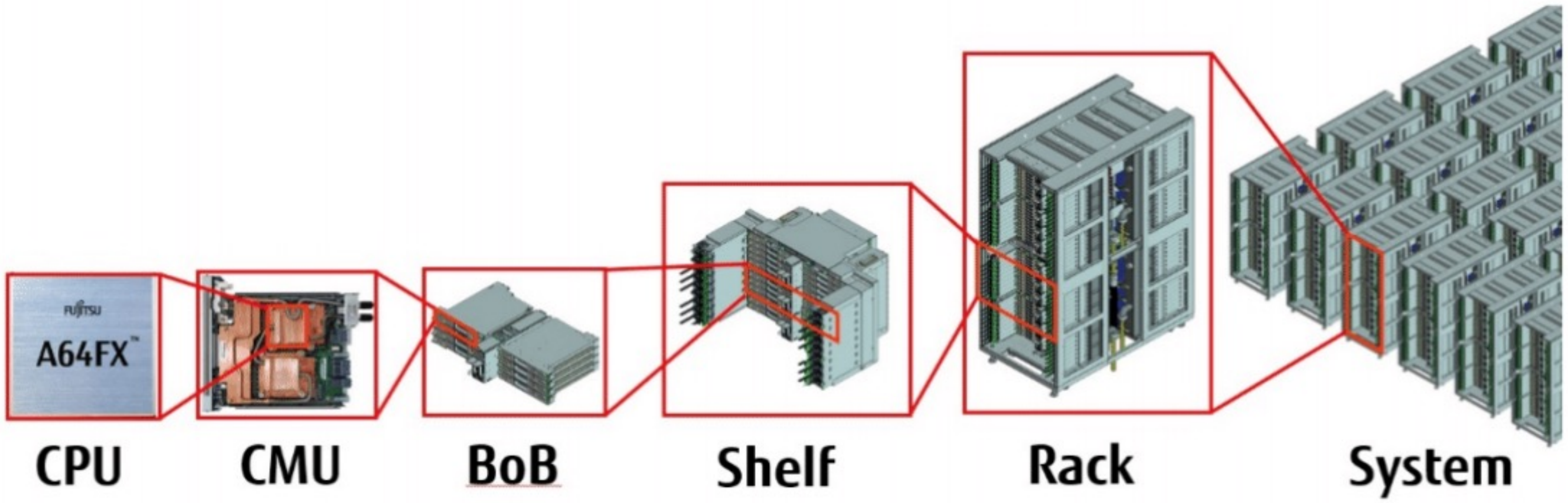
UNIVERSITY
OF TWENTE.

PUTTING IT ALL TOGETHER: IBM'S BLUEGENE/Q



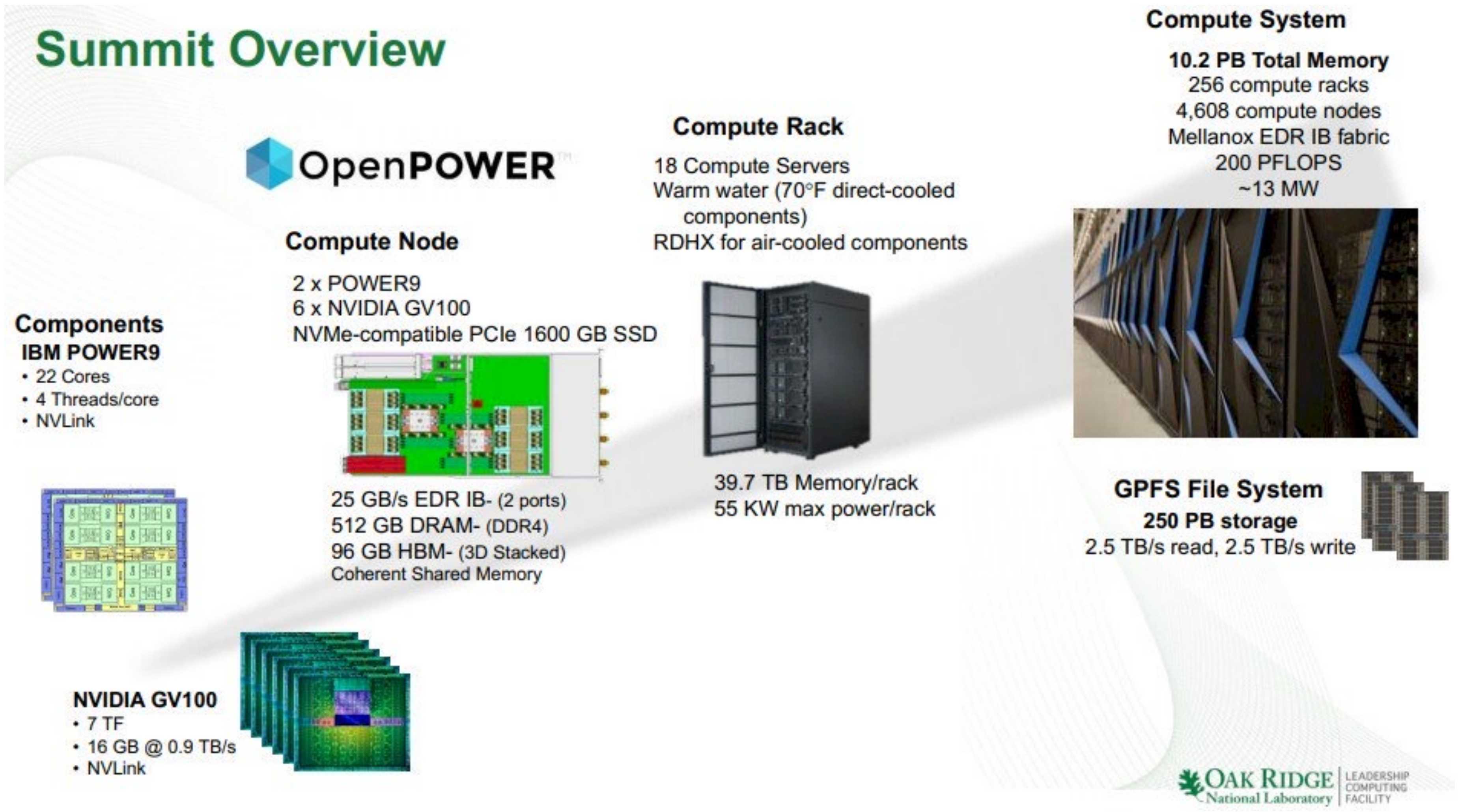
- **Sustained single node perf: 10x P, 20x L**
- **MF/Watt: (6x) P, (10x) L (~2GF/W, Green 500 criteria)**
- **Software and hardware support for programming models for exploitation of node hardware concurrency**

PUTTING IT ALL TOGETHER: FUGAKU



UNIVERSITY
OF TWENTE.

PUTTING IT ALL TOGETHER: SUMMIT



Summit Overview

OpenPOWER™

Components

IBM POWER9
- 22 Cores
- 4 Threads/core
- NVLink

NVIDIA GV100
- 7 TF
- 16 GB @ 0.9 TB/s
- NVLink

Compute Node

2 x POWER9
6 x NVIDIA GV100
NVMe-compatible PCIe 1600 GB SSD

25 GB/s EDR IB- (2 ports)
512 GB DRAM- (DDR4)
96 GB HBM- (3D Stacked)
Coherent Shared Memory

Compute Rack

18 Compute Servers
Warm water (70°F direct-cooled components)
RDHX for air-cooled components

39.7 TB Memory/rack
55 KW max power/rack

Compute System

10.2 PB Total Memory
256 compute racks
4,608 compute nodes
Mellanox EDR IB fabric
200 PFLOPS
~13 MW

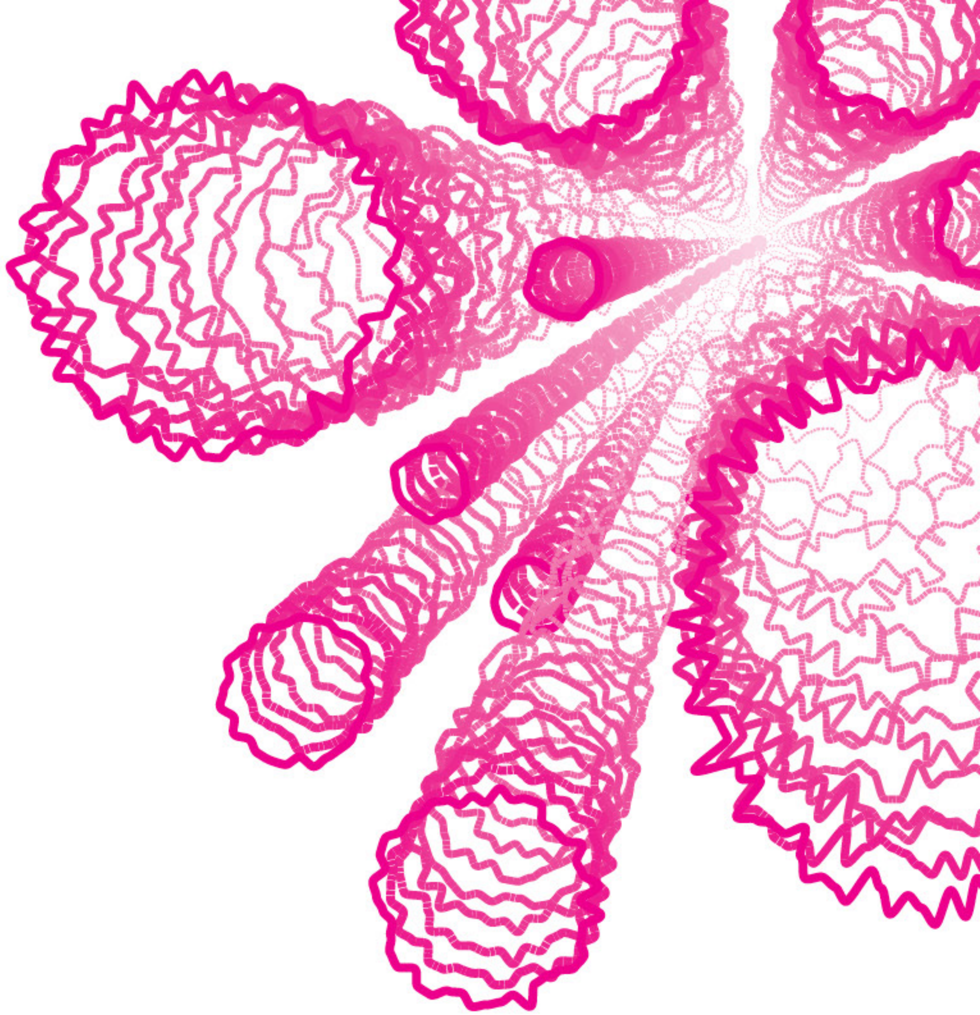
GPFS File System

250 PB storage
2.5 TB/s read, 2.5 TB/s write

OAK RIDGE National Laboratory | LEADERSHIP COMPUTING FACILITY

UNIVERSITY OF TWENTE.

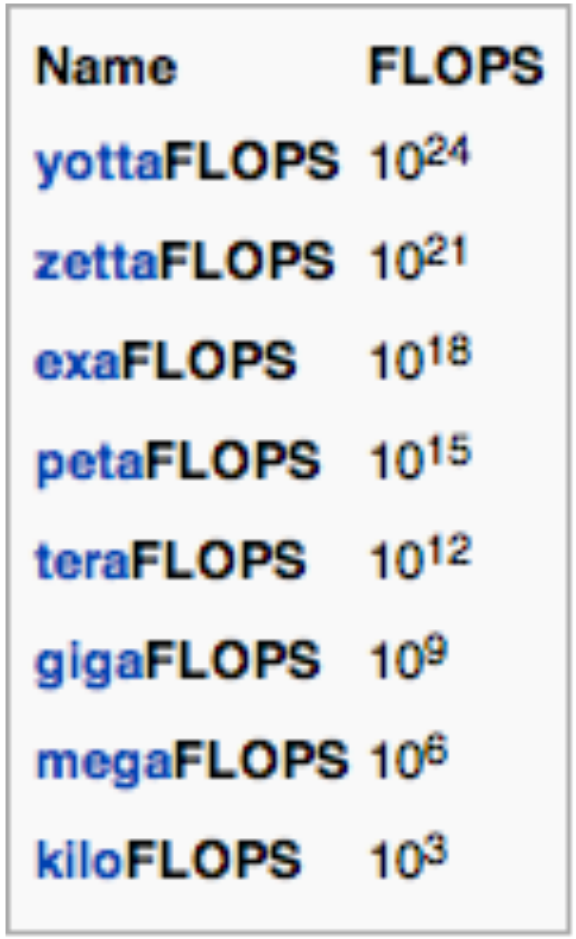
HARDWARE PERFORMANCE



UNIVERSITY OF TWENTE.

PERFORMANCE “METRICS”

- Clock frequency [GHz] = absolute hardware speed
  - Memories, CPUs, interconnects
- **Operational speed [GFLOPs]**
  - Operations per second, **single/double/…** precision
- **Memory bandwidth [GB/s]**
  - Memory operations per second
  - Differs a lot between different memories on chip
- Derived metrics
  - FLOP/Byte, FLOP/Watt



Name	FLOPS
yottaFLOPS	10^{24}
zettaFLOPS	10^{21}
exaFLOPS	10^{18}
petaFLOPS	10^{15}
teraFLOPS	10^{12}
gigaFLOPS	10^{9}
megaFLOPS	10^{6}
kiloFLOPS	10^{3}

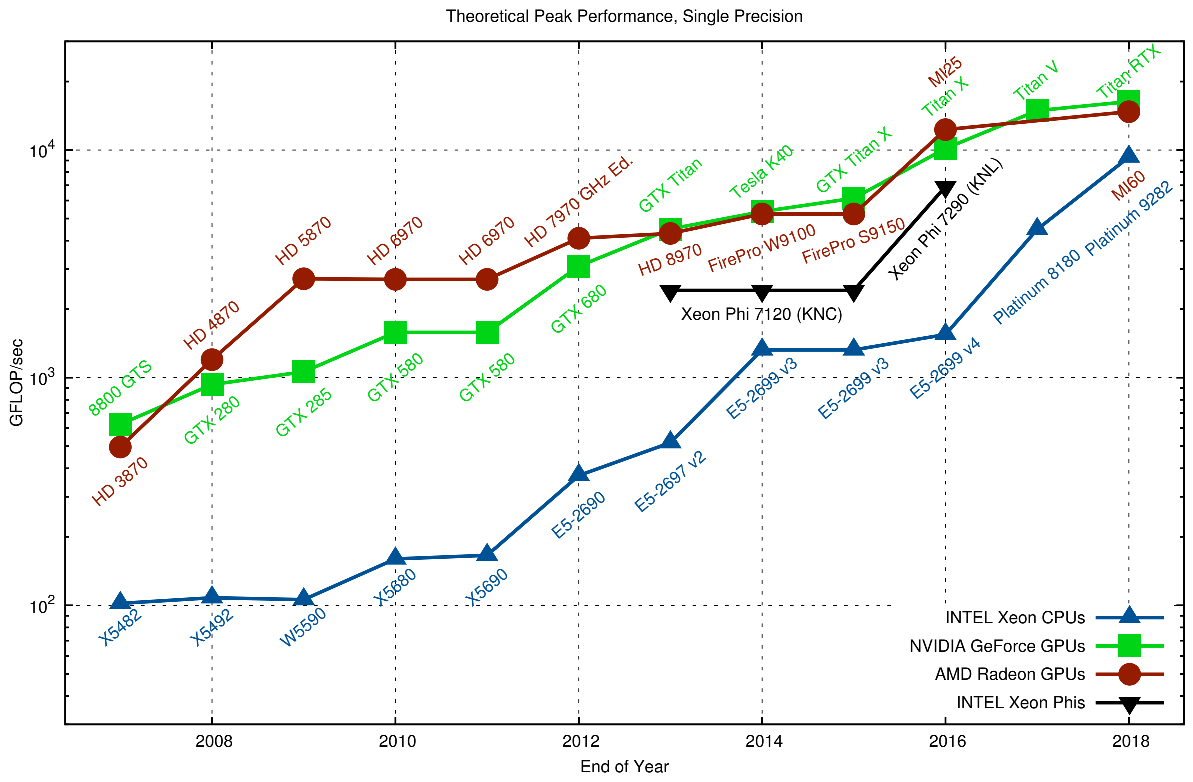
THEORETICAL PEAK PERFORMANCE

```
Throughput[GFLOP/s] = chips * cores * vectorWidth *
                                  FLOPs/cycle * clockFrequency
Bandwidth[GB/s] = memory bus frequency * bits per cycle * bus width
```

	Cores	Threads/ALUs	Throughput	Bandwidth
Intel Core i7	4	16	85	25.6
AMD Barcelona	4	8	37	21.4
AMD Istanbul	6	6	62.4	25.6
NVIDIA GTX 580	16	512	1581	192
NVIDIA GTX 680	8	1536	3090	192
AMD HD 6970	384	1536	2703	176
AMD HD 7970	32	2048	3789	264
Intel Xeon Phi 7120	61	240	2417	352

UNIVERSITY OF TWENTE.

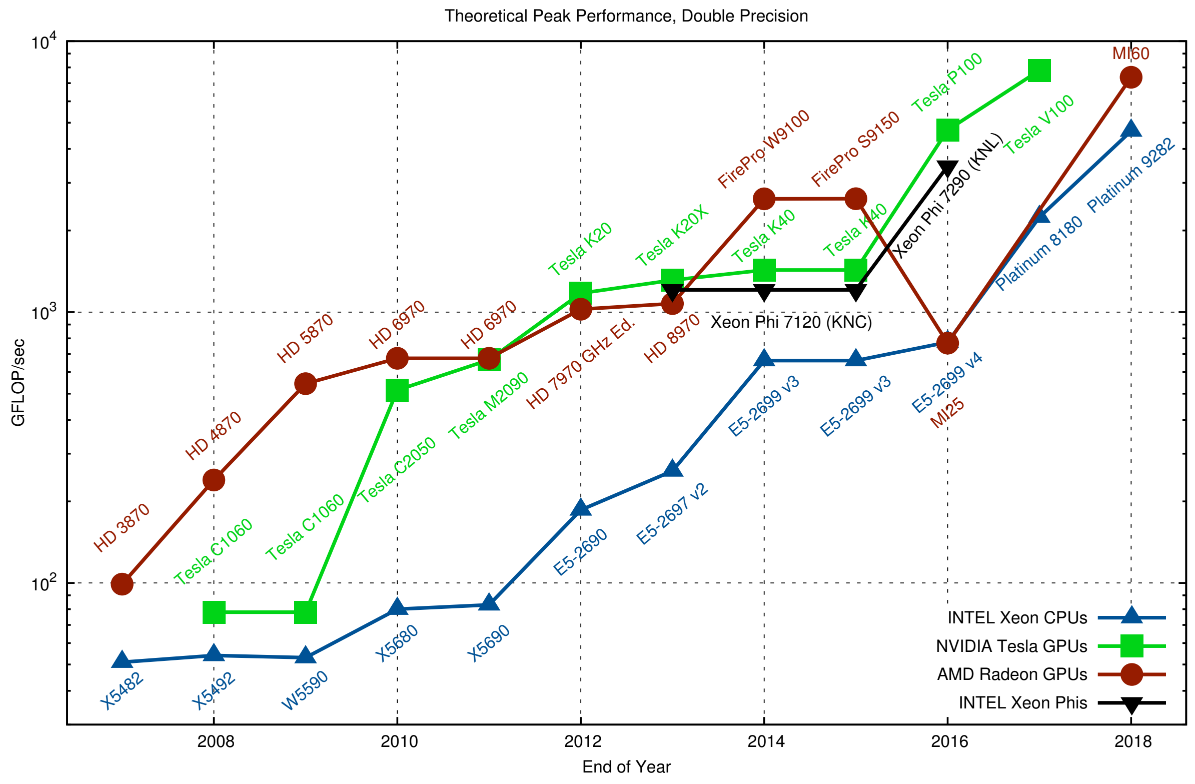
MULTI VS *MANY* CORES (SP-FLOPS)



UNIVERSITY OF TWENTE.

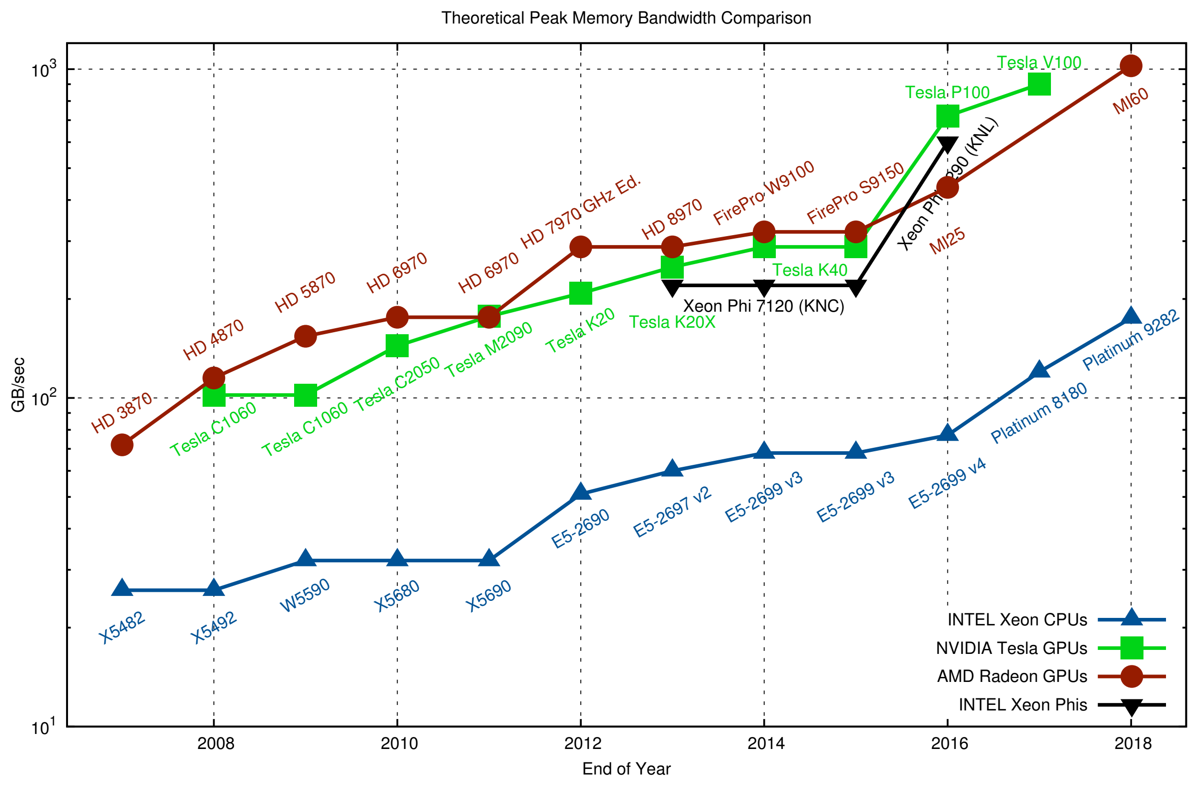
https://www.karlrupp.net/2013/06/cpu-gpu-and-mic-hardware-characteristics-over-time/

MULTI VS *MANY* CORES (DP-FLOPS)



https://www.karlrupp.net/2013/06/cpu-gpu-and-mic-hardware-characteristics-over-time/

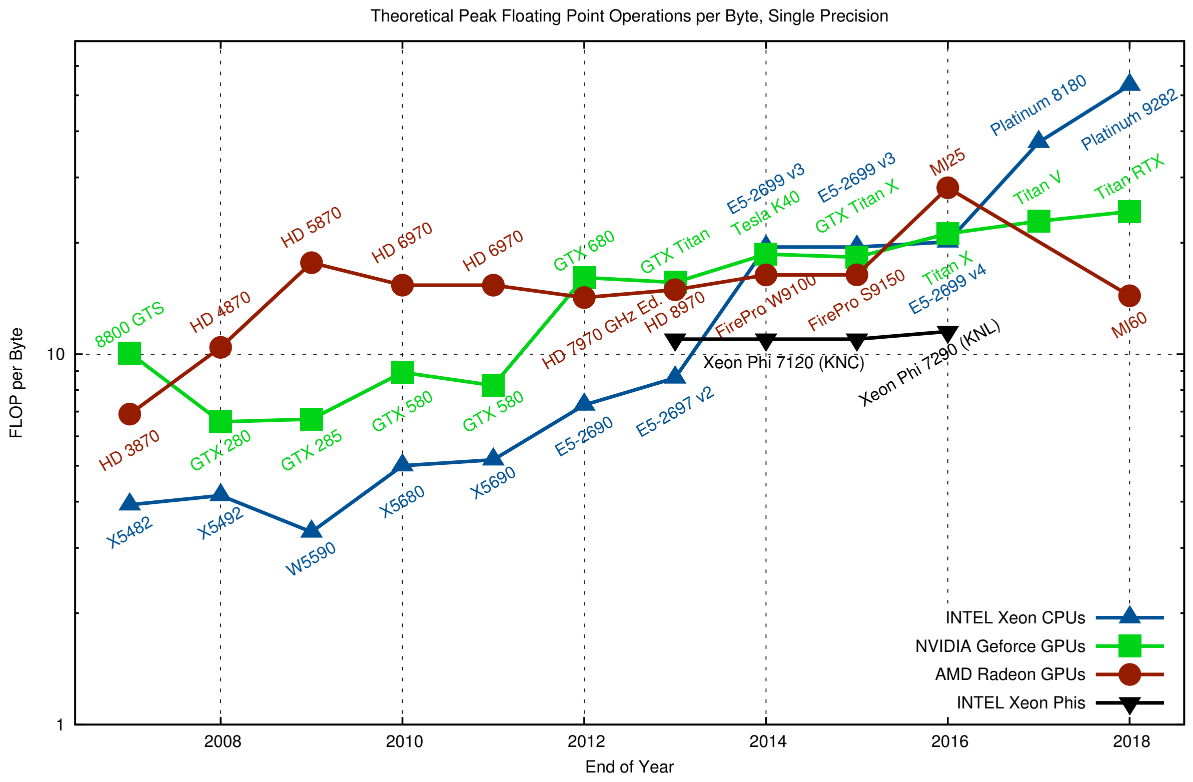
MULTI VS *MANY* CORES (GB/S)



UNIVERSITY OF TWENTE.

https://www.karlrupp.net/2013/06/cpu-gpu-and-mic-hardware-characteristics-over-time/

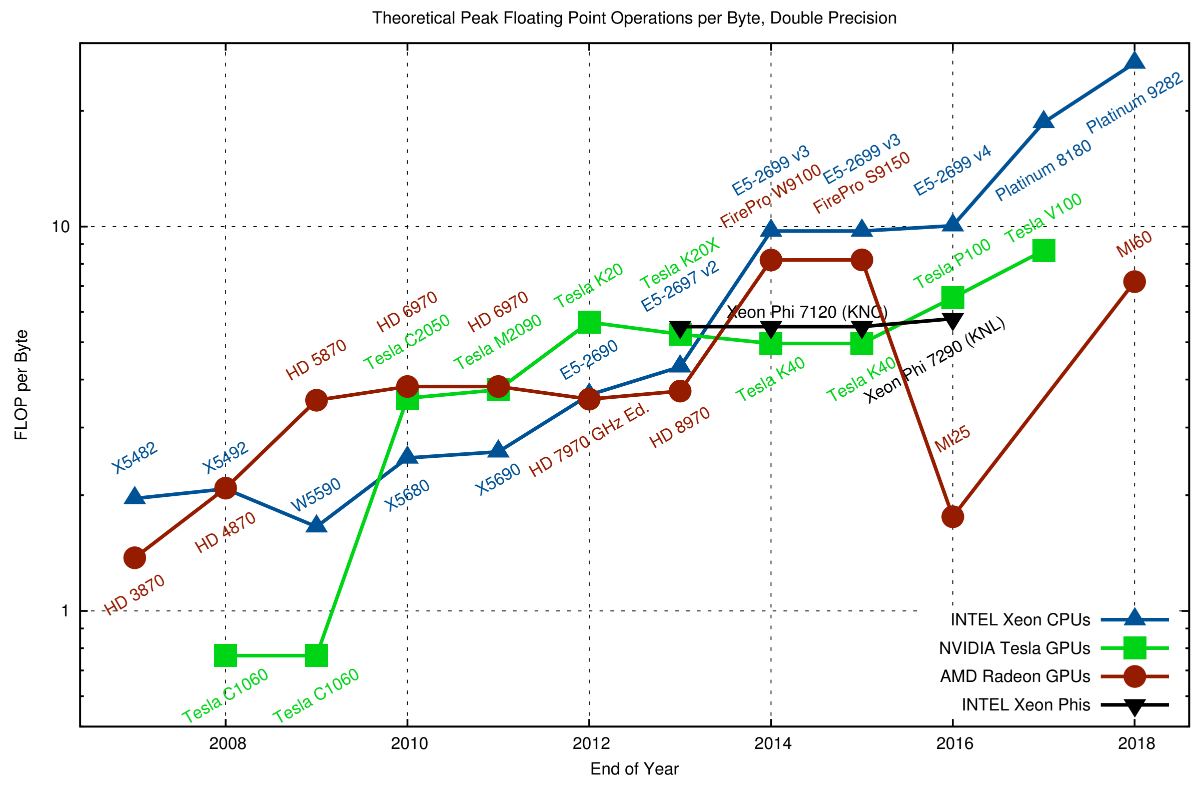
BALANCE ? FLOPs/Byte (SP) !



UNIVERSITY OF TWENTE.

https://www.karlrupp.net/2013/06/cpu-gpu-and-mic-hardware-characteristics-over-time/

BALANCE ? FLOPs/Byte (DP) !



UNIVERSITY OF TWENTE.

https://www.karlrupp.net/2013/06/cpu-gpu-and-mic-hardware-characteristics-over-time/

WHY SHOULD WE CARE?

- Peak performance indicates an absolute bound of the performance that can be achieved on a given machine
  - It is *application independent*

- Such performance is rarely* achievable in practice for real applications.
  - Applications rarely utilize all the machine features.

- The balance of an application must *consistently* match the balance of the machine to get anywhere near the peak…
- ... or else… different bottlenecks!

*Empirical studies show this reads as “almost never” .

https://gitlab.com/astron-misc/benchmark-intrinsics/-/tree/master

UNIVERSITY OF TWENTE.

ABSOLUTE HARDWARE PERFORMANCE

- Only achieved in the optimal conditions:
  - Processing units 100% used
  - All parallelism 100% exploited
  - All data transfers at maximum bandwidth

- In real life
  - No application is like this
  - Can we reason about “real” performance?

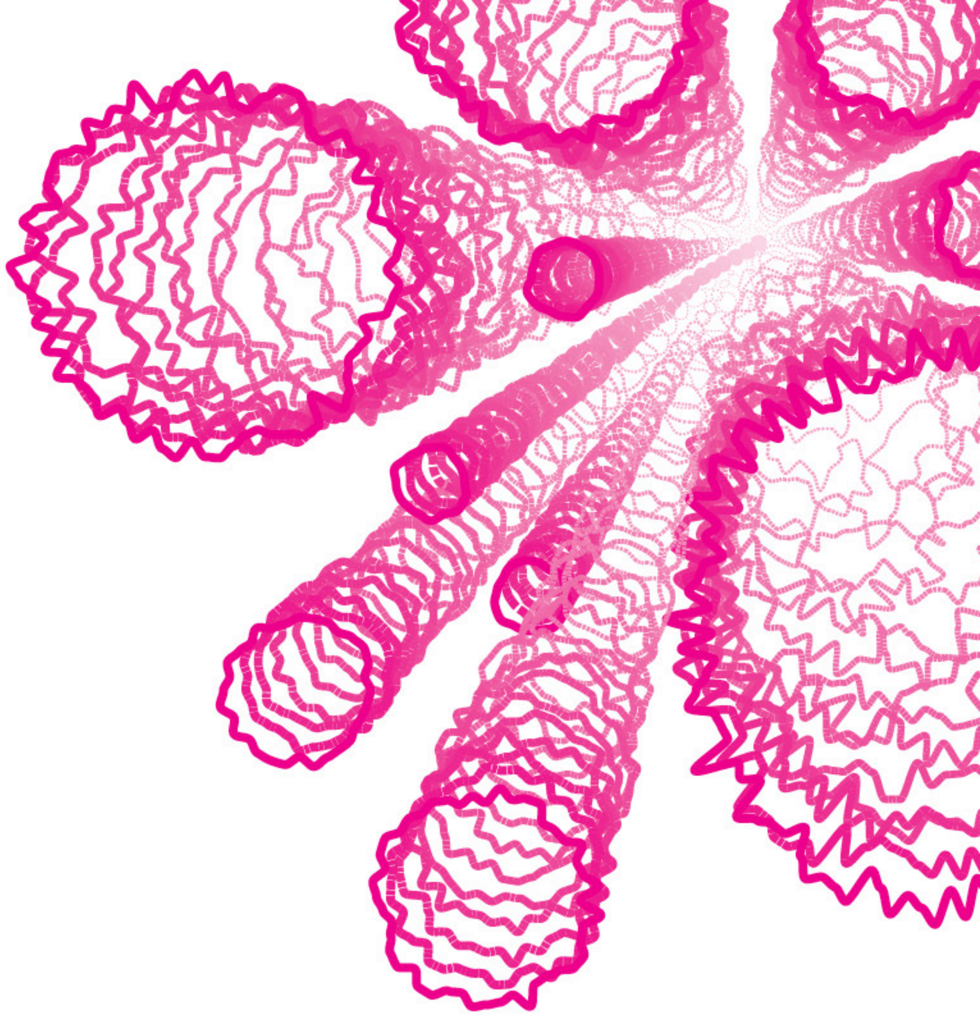
“REAL” HARDWARE PERFORMANCE

- Microbenchmarking*
  - Evaluates hardware features in isolation
  - Goal: find out the true limits of the hardware components
  - Platform-specific results
  - **Compared** with the theoretical peak, **per platform**.

- Benchmarking
  - Evaluates the FULL platform
  - Application-specific performance
    - Top500 – computation capability
    - Graph500 – graph processing capability
    - Green500 – energy consumption
  - **Compares platforms**

* Henry Wong, Misel-Myrto Papadopoulou, Maryam Sadooghi-Alvandi, Andreas Moshovos. ”Demystifying GPU Microarchitecture throu

WHAT IF YOU WANT TO KNOW MORE?



UNIVERSITY OF TWENTE.

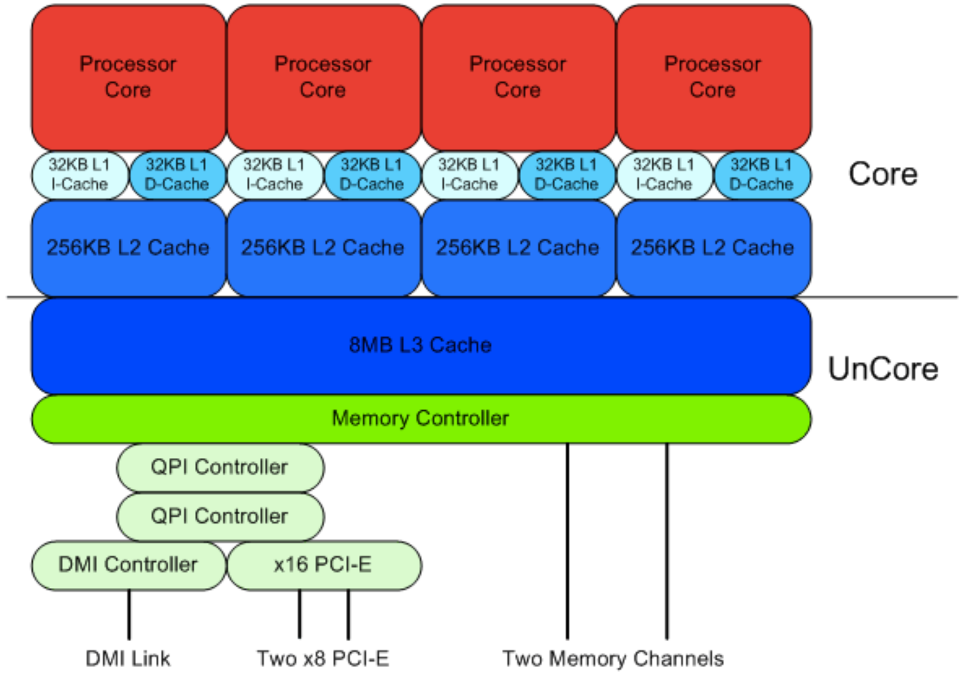
MEET HARDWARE PERFORMANCE COUNTERS

- A set of special-purpose registers built into modern microprocessors

- Store the counts of hardware-related activities/events

- Counters = the actual registers
- Events = actual hardware events
  - Events / counters >> 1 => reprogramming the counters !

- Performance Monitoring Units: hardware units to monitor performance
  - Core: what happens at core level
  - Uncore: outside the core



TYPES OF COUNTERS (EXAMPLES)

Core-events

- instructions retired
- elapsed core clock ticks
- core frequency
- memory subsystem (L1, L2)

UnCore-events

- LLC
- Read/written bytes from/to memory controller(s)
- Data traffic transferred by the QPI links.

Literally hundreds and hundreds more

UNIVERSITY
OF TWENTE.

WARNINGS : COMPLEXITY

- High-complexity of hardware => many different types of counters
  - It is rarely the case that a single event tells a complete story

- Intel splits events in *architectural* and *non-architectural*
  - i.e., processor independent vs. processor dependent

- Different generations => different *non-architectural* counters
  - Different names
  - Different meanings

- Typically used to confirm/infirm hypotheses
  - A counter on its own won't tell you much if you don't have any expectation …

TOOLS & METHODS

- Low-level assembly code
  - Set what needs to be counted …
  - … and keep reading the register!

- PAPI (http://icl.cs.utk.edu/papi/overview/index.html)
  - Portable interface across devices
  - Simple API to access most counters &

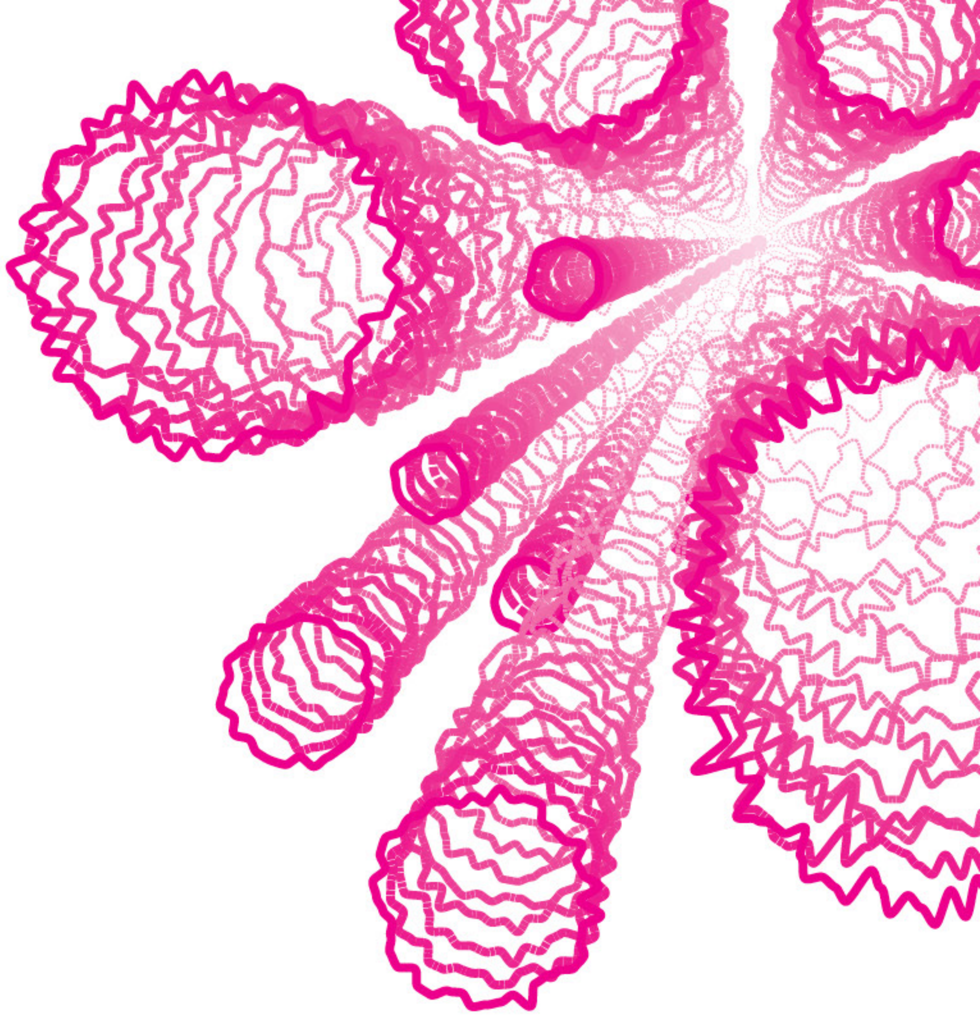
- High-level tools
  - Intel VTune
  - AMD uProf
  - NVIDIA nsight/nvprof
  - **LIKWID**
    - Lots of tools to simplify collection

```
int event[NUM_EV]={PAPI_TOT_INS, PAPI_TOT_CYC, PAPI_L1_DCM };
long long values[NUM_EV];

/* Start counting events */
PAPI_start_counters(event, NUM_EV);
//call function
PAPI_read_counters(values, NUM_EV);

printf("Total instructions: %lld\n", values[0]);
/* Stop counting events */
PAPI_stop_counters(values, NUM_EVENTS)
```

HPC PULSE



UNIVERSITY
OF TWENTE.

HPC PULSE

- TOP500 Project*
  - The 500 most powerful computers in the world
- Benchmark: Rmax of LINPACK
  - Solve the Ax=b linear system
    - dense problem
    - matrix A is random
  - Dominated by dense matrix-matrix multiply
- Metric: FLOPS/s
  - Computational throughput: number of floating point operations per second
- Updated twice a year: latest is June 2023

*Read more: www.top500.org

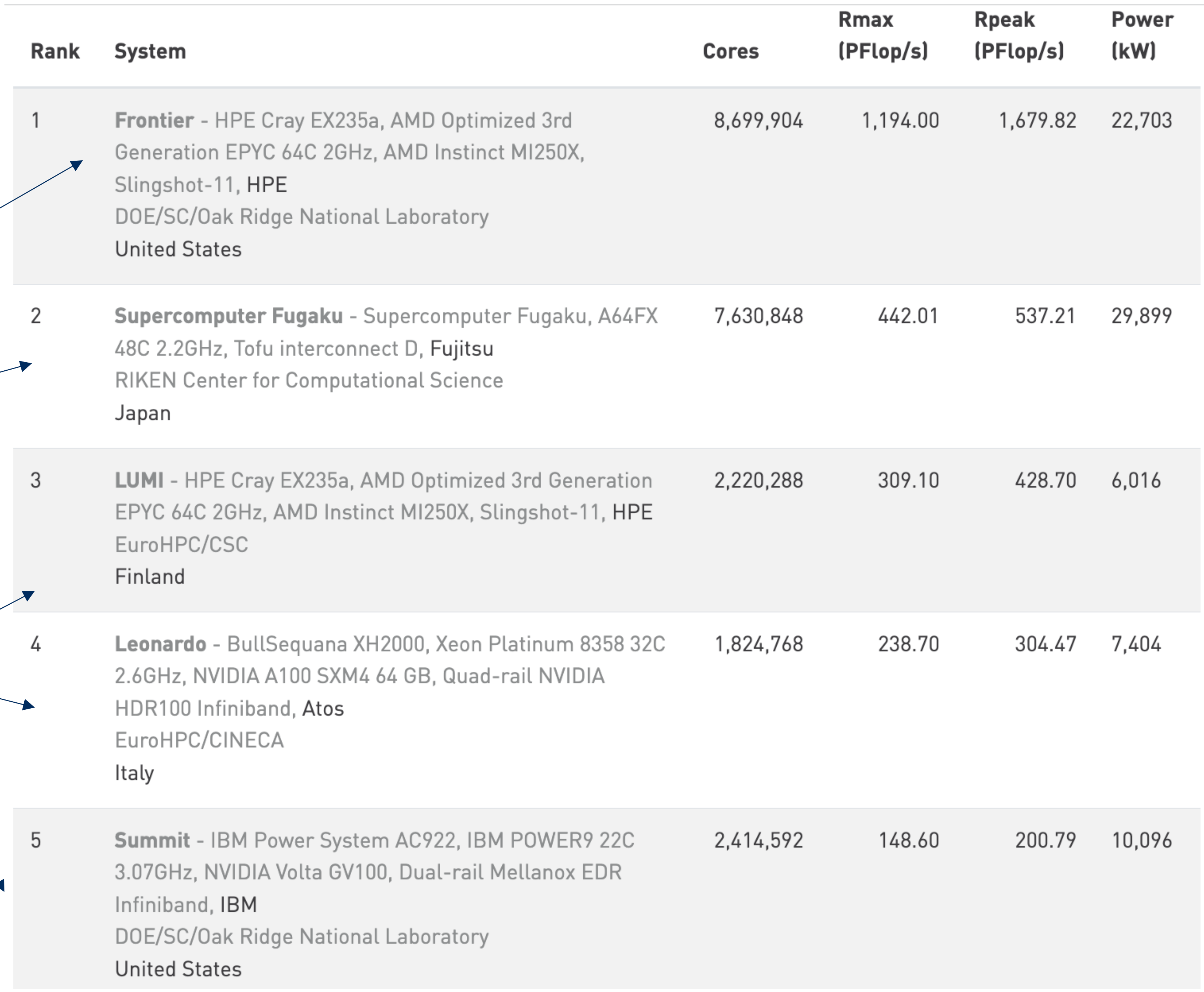
UNIVERSITY OF TWENTE.

TOP5(00) JUNE 2023

1 Exascale machine

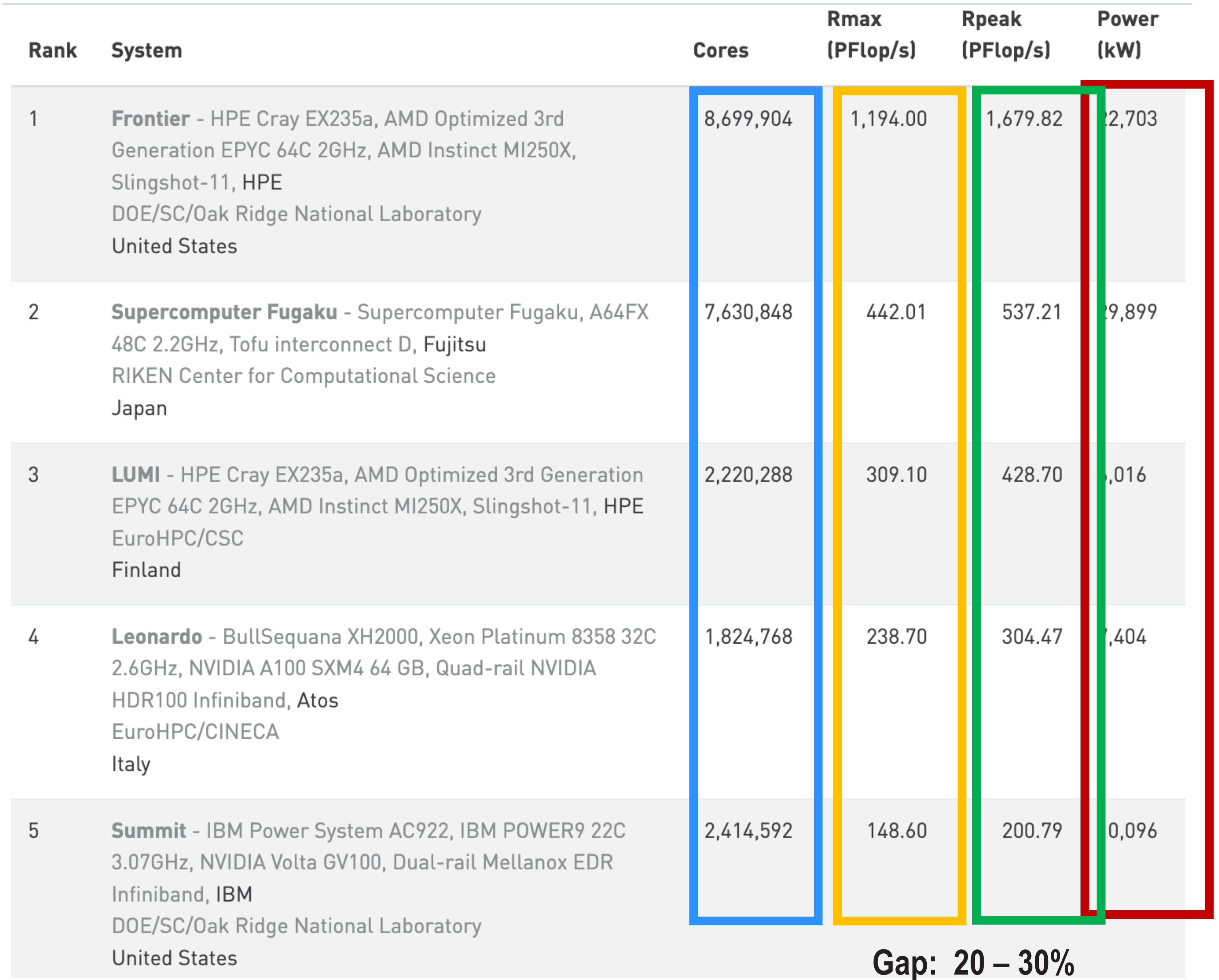
1 custom-built machine

3 energy-efficient machines: AMD, Intel+NVIDIA, IBM



Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,699,904	1,194.00	1,679.82	22,703
2	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
3	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,220,288	309.10	428.70	6,016
4	**Leonardo** - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, Atos EuroHPC/CINECA Italy	1,824,768	238.70	304.47	7,404
5	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148.60	200.79	10,096

TOP500 JUNE 2023



Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,699,904	1,194.00	1,679.82	2,703
2	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	9,899
3	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,220,288	309.10	428.70	,016
4	**Leonardo** - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, Atos EuroHPC/CINECA Italy	1,824,768	238.70	304.47	,404
5	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148.60	200.79	0,096

Gap: 20 – 30%

Lots of **accelerators**.

Many many **many cores**.

High **peak** performance.

Large **gap** from peak to "real" performance.

Let's talk about energy.

UNIVERSITY OF TWENTE.

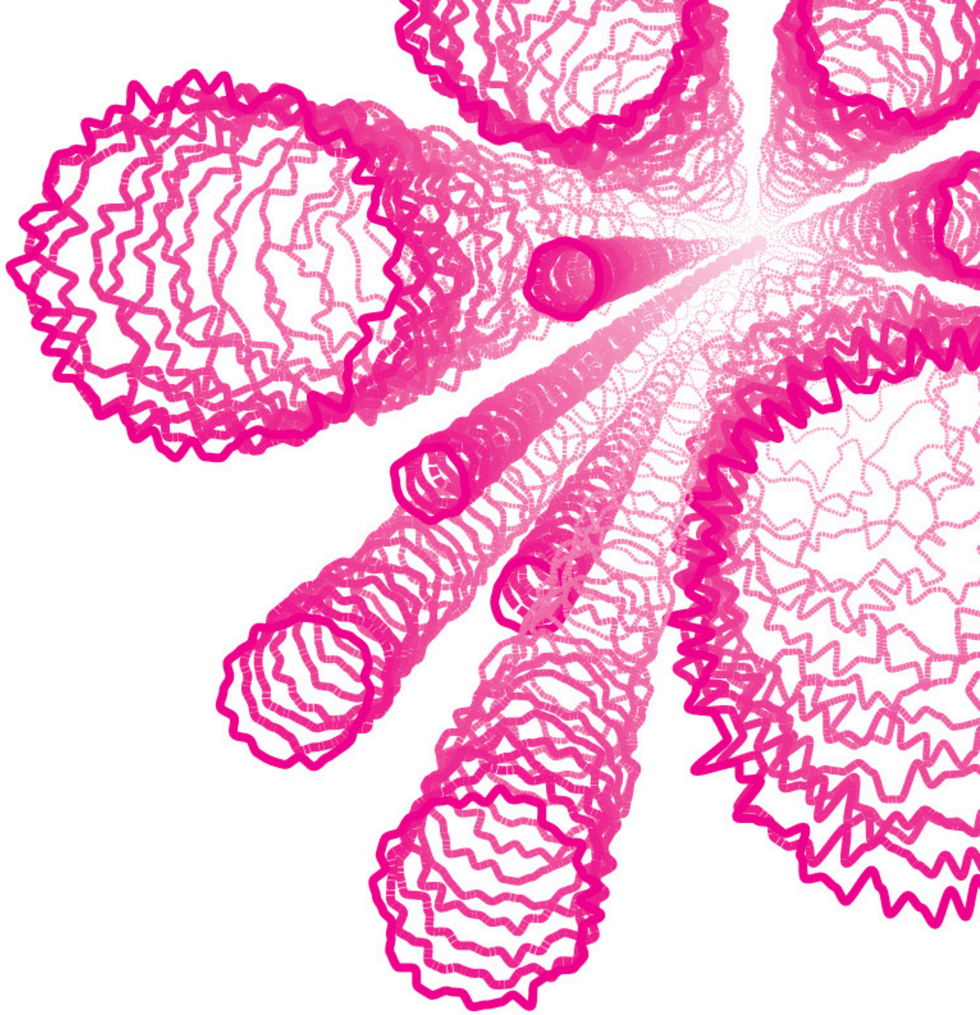
See more at: https://www.top500.org/

GREEN 500

Rank	TOP500 Rank	System	Cores	Rmax (PFlop/s)	Power (kW)	Energy Efficiency (GFlops/watts)
1	255	Henri - ThinkSystem SR670 V2, Intel Xeon Platinum 8362 32C 2.8GHz, NVIDIA H100 80GB PCIe, Infiniband HDR, Lenovo United States	8,288	2.88	44	65.396
2	34	Frontier TDS - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE United States	120,832	19.20	309	62.684
3	12	Adastra - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE France	319,072	46.10	921	58.021
4	17	Setonix – GPU - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE Australia	181,248	27.16	477	56.983
5	77	Dardel GPU - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE Sweden	52,864	8.26	146	56.491

UNIVERSITY
OF TWENTE.

IN SUMMARY …



UNIVERSITY
OF TWENTE.

TODAY’S COMPUTING MACHINES

- Parallel at different levels
  - Multi- or many-cores
  - Core-level parallelism
  - CPU-accelerator(s) parallelism
  - None-level parallelism
- Different performance and power “profiles” => different energy consumption => different energy efficiency envelops
- Hardware-level performance
  - FLOPs (or INTOPs) for computation
  - GB/s for memory bandwidth
  - FLOPS/Watt for energy efficiency
- Benchmarking machine performance
  - Micro-benchmarking vs. benchmarking
  - Diverse metrics => “performance counters”