Outline

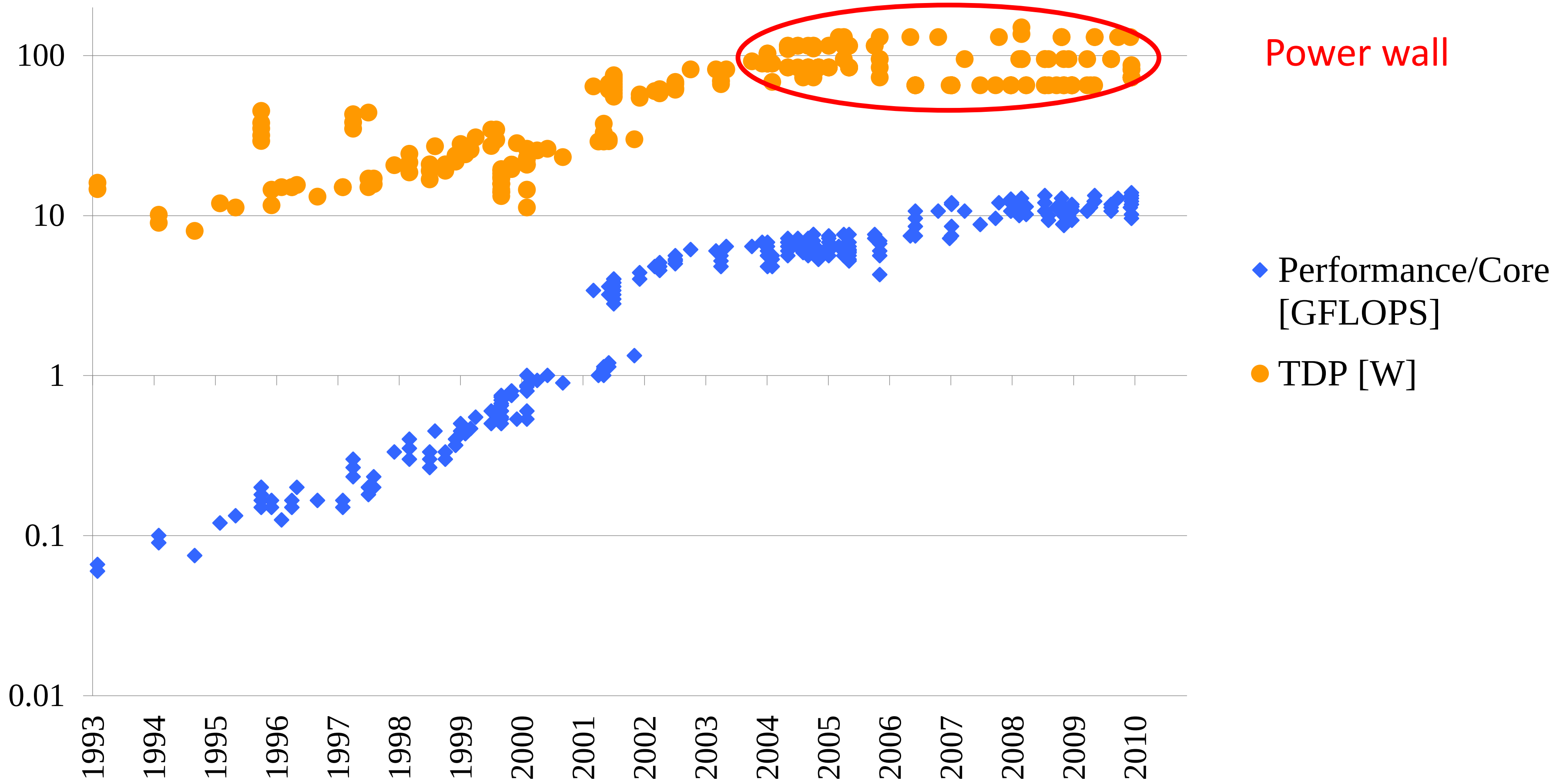
1. Why supercomputers?
2. Modern processors
3. Basic optimization techniques for serial code
4. Data access optimization
5. Parallel computers
6. Parallelization and optimization strategies
7. Shared-memory programming with OpenMP
8. Distributed-memory programming with MPI
9. Hybrid programming (MPI + OpenMP)
10. Parallel algorithms
11. Heterogeneous architectures (GPUs, Xeon Phis)

12. **Energy efficiency**
    → **Motivation**
        → **Towards recent HPC systems**
        → Trends
    → A (very short) side trip into physics
    → Energy efficiency of microarchitectures
    → Benchmarking
    → Outlook

Single Core Peak Performance and Processor TDP

- **TDP = thermal design power**

  → max. amount of heat generated by the CPU in typical operation



TOP500 List November 2011: Nr. 1 and Nr. 2 Systems

- **K-Computer at RIKEN (Japan)**

→ No. 1 system

→ SPARC64 VIIIfx 2.0 GHz

→ **12.6 MW**

→ 830 MFlops/Watt

- **Tianhe-1A (China)**

→ No. 2 system

→ Intel X5670 + NVIDIA GPU

→ **4.04 MW**

→ 635 MFlops/Watt

TOP500 List November 2012: Nr. 1 and Nr. 2 Systems

- Cray Titan

→ Nr. 1 System

→ Cray XK7: Opteron + Tesla K20x

→ **8.2 MW**

→ 2142 MFlops/Watt

- IBM Blue Gene/Q Sequoia at LLNL

→ No. 2 system

→ POWER BQC 16C 1.60 GHz

→ **7.8 MW**

→ 2069 MFlops/Watt

TOP500 List November 2015:
Nr. 1 and Nr. 2 Systems

Tianhe-2 (MilkyWay-2)

- → Nr. 1 System
- → TH-IVB-FEP Cluster
- → 2 Xeon Ivy Bridge + 3 Xeon Phi
- → **17.8 MW**

Cray Titan

- → Nr. 2 System
- → Cray XK7: Opteron + Tesla K20x
- → **8.2 MW**
- → 2142 MFlops/Watt

TOP500 List November 2016: Nr. 1 and Nr. 2 Systems

- **Sunway TaihuLight**

→ Nr. 1 System

→ Sunway MPP

→ Sunway SW26010

→ **15.4 MW**

- **Tianhe-2 (MilkyWay-2)**

→ Nr. 2 System

→ TH-IVB-FEP Cluster

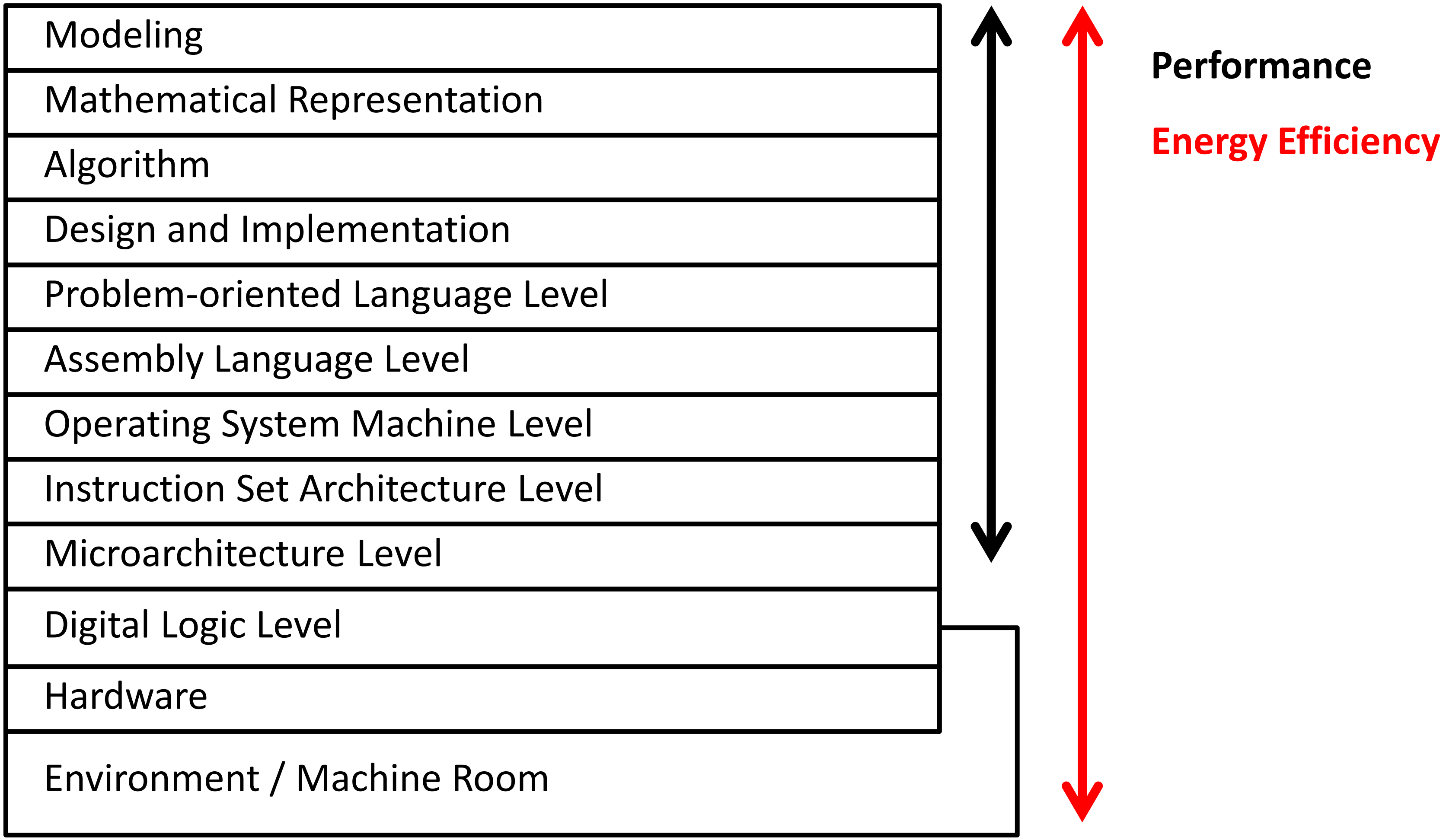
→ 2 Xeon Ivy Bridge + 3 Xeon Phi

→ **17.8 MW**

Today‘s supercomputers need between 4-18 MW.

Why is energy efficiency hard?

- **Energy efficiency is a vertical/cross cutting problem**



Modeling

Mathematical Representation

Algorithm

Design and Implementation

Problem-oriented Language Level

Assembly Language Level

Operating System Machine Level

Instruction Set Architecture Level

Microarchitecture Level

Digital Logic Level

Hardware

Environment / Machine Room

Performance

Energy Efficiency

Outline

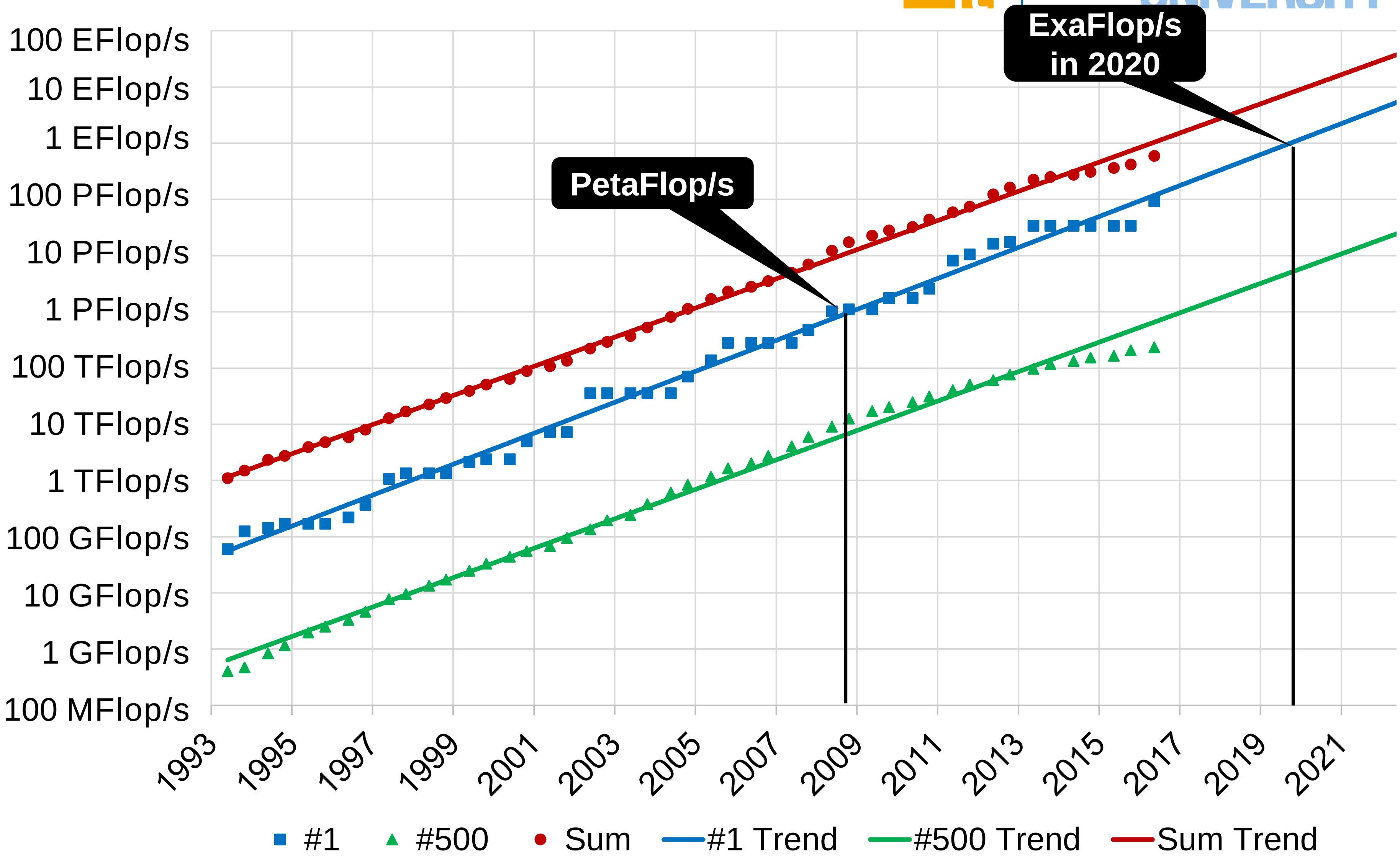
1. Why supercomputers?
2. Modern processors
3. Basic optimization techniques for serial code
4. Data access optimization
5. Parallel computers
6. Parallelization and optimization strategies
7. Shared-memory programming with OpenMP
8. Distributed-memory programming with MPI
9. Hybrid programming (MPI + OpenMP)
10. Parallel algorithms
11. Heterogeneous architectures (GPUs, Xeon Phis)

12. **Energy efficiency**
    - → **Motivation**
        - → Towards recent HPC systems
        - → **Trends**
    - → A (very short) side trip into physics
    - → Energy efficiency of microarchitectures
    - → Benchmarking
    - → Outlook

Trends

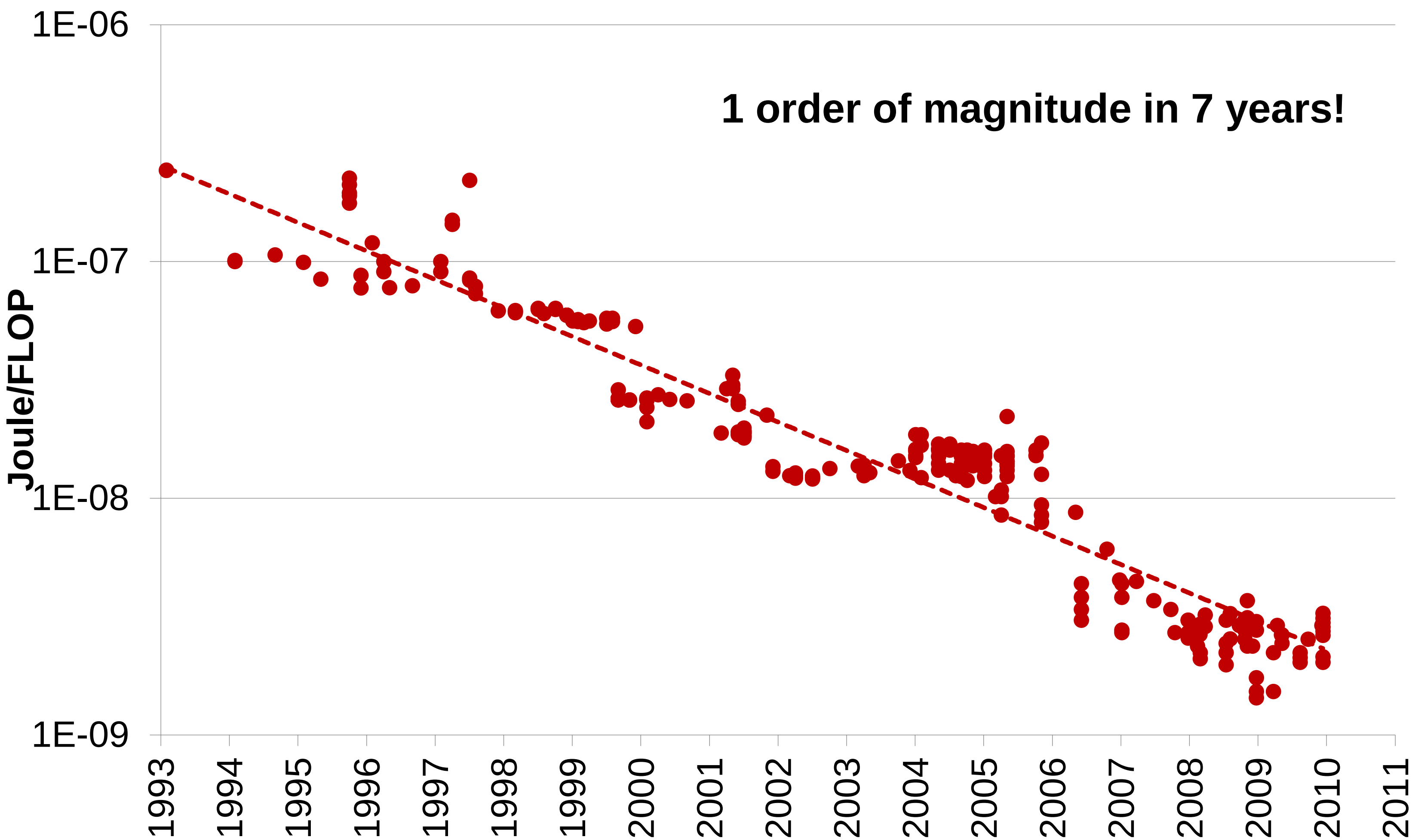
- **TOP500 list (top500.org):**
  - → Good indicator for the performance development of HPC systems
  - → Dates back to 1993
  - → No real power measurement included
    - →Analyze trends in performance & power consumption

Projected performance development in TOP500

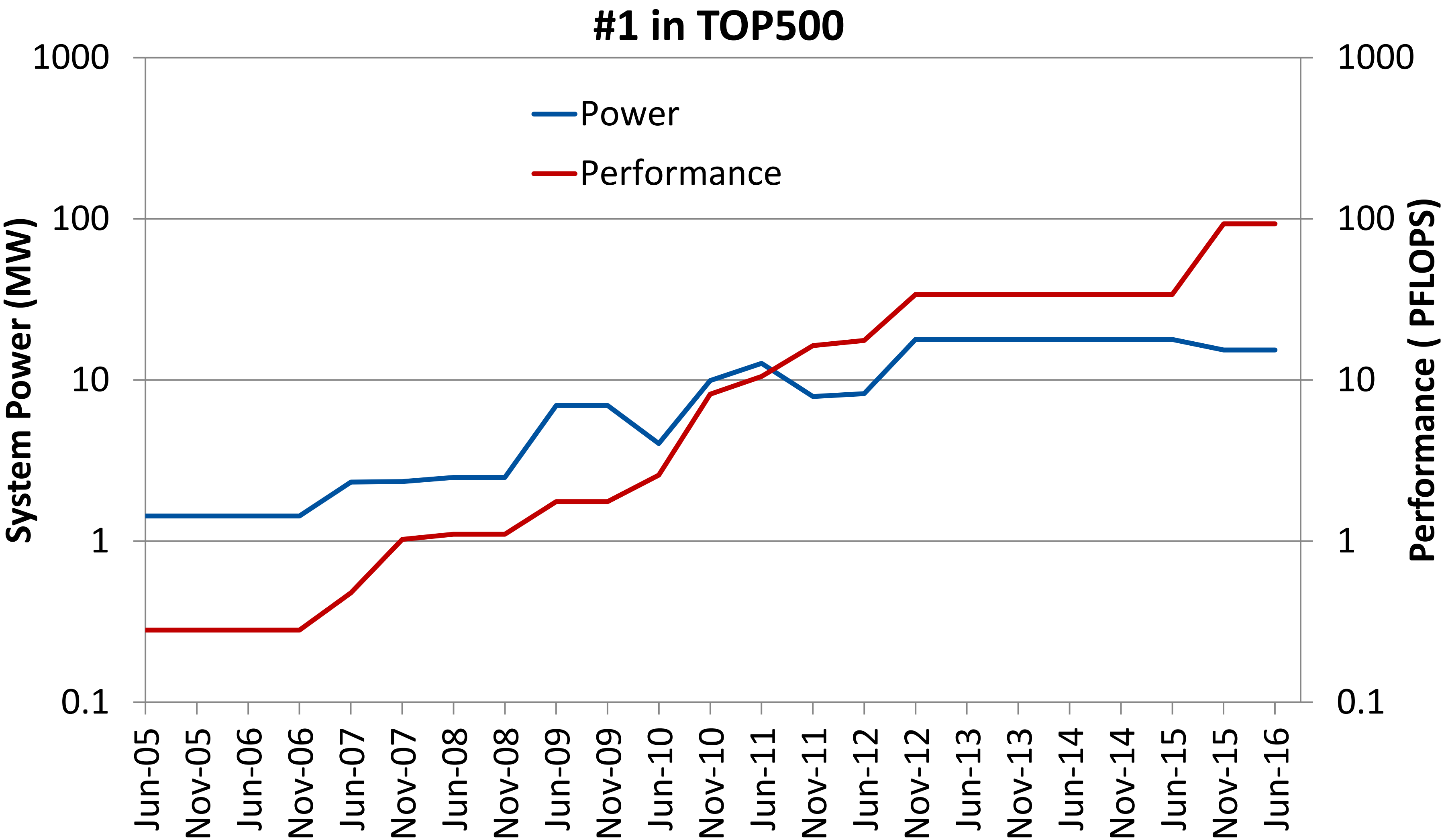


Source:Top500 06/2016

Energy Efficiency of Intel CPUs (TDP/Peak Performance)

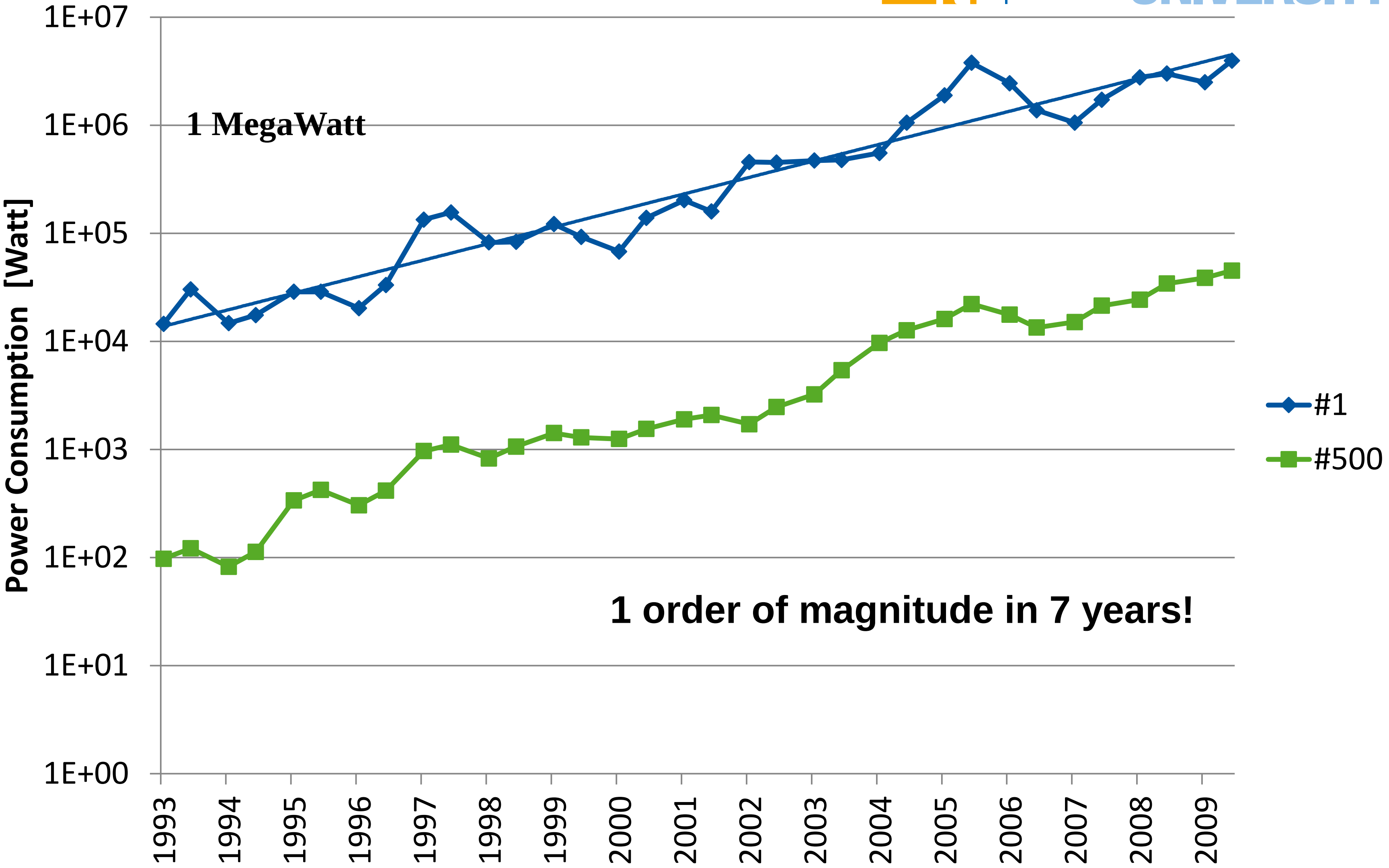


Motivation – HPC
Power Consumption and Performance



Source:Top500 11/2016

Estimated Energy Consumption of TOP500 systems



1 order of magnitude in 7 years!

Trends

- Performance increase: 2 orders of magnitude in 7 years
- Energy efficiency decrease: 1 order of magnitude in 7 years

→ $energy\ efficiency = \frac{energy}{FLOP} = \frac{power\ consumption \cdot time}{FLOP} = \frac{power\ consumption}{Flop/s}$

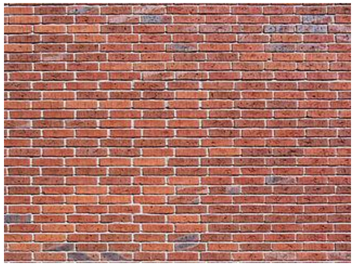
→ $power\ consumption = energy\ efficiency \cdot Flop/s$

- Power consumption increase: 1 order of magnitude in 7 years

Where are we heading?

- **PetaFLOP reached in 2008**
- **Next big goal: ExaFLOP in 2020**
- **But: Current power consumption trend cannot continue**

power wall

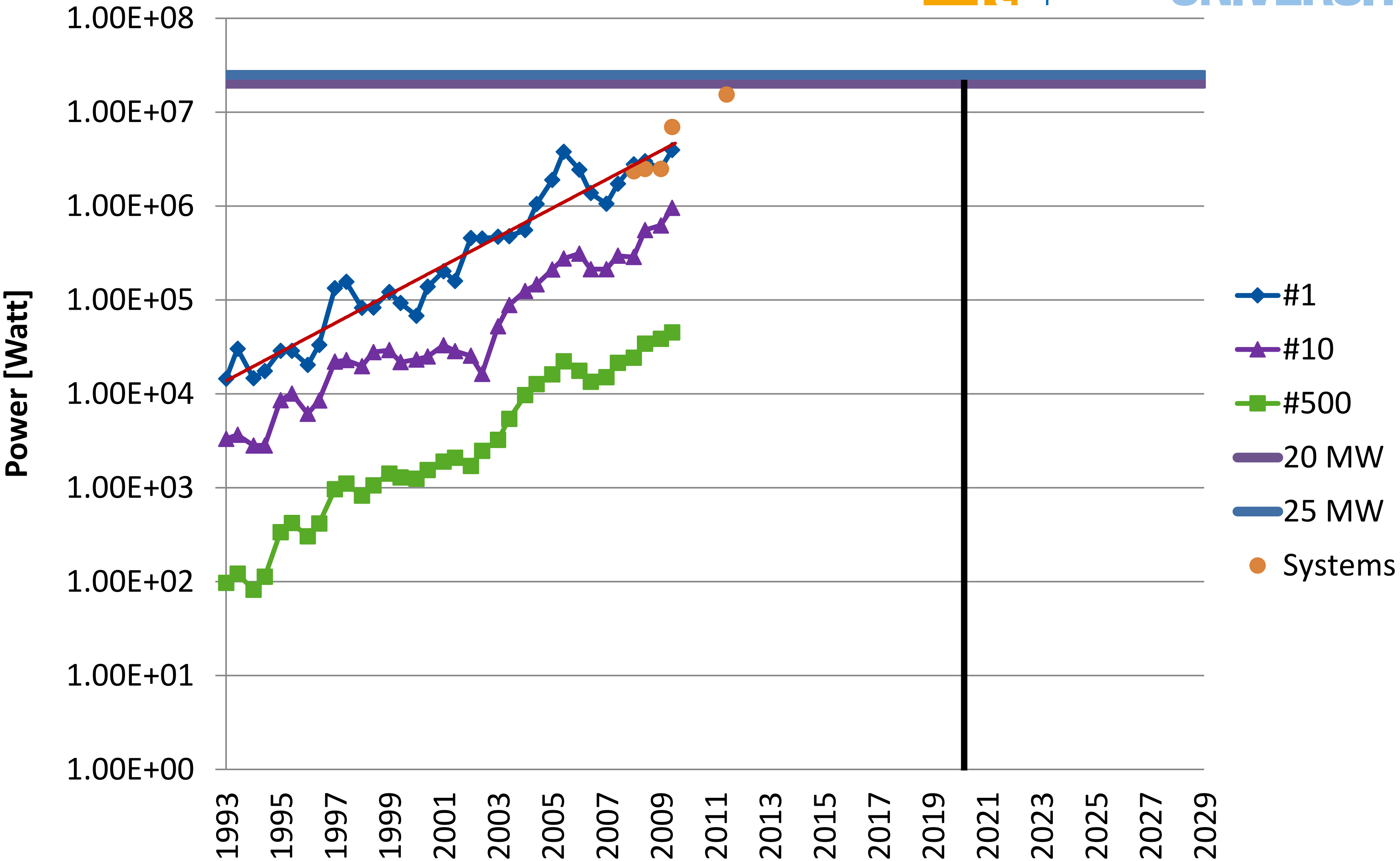


- **20 to 25 MegaWatt considered to be an important barrier[1,2]**

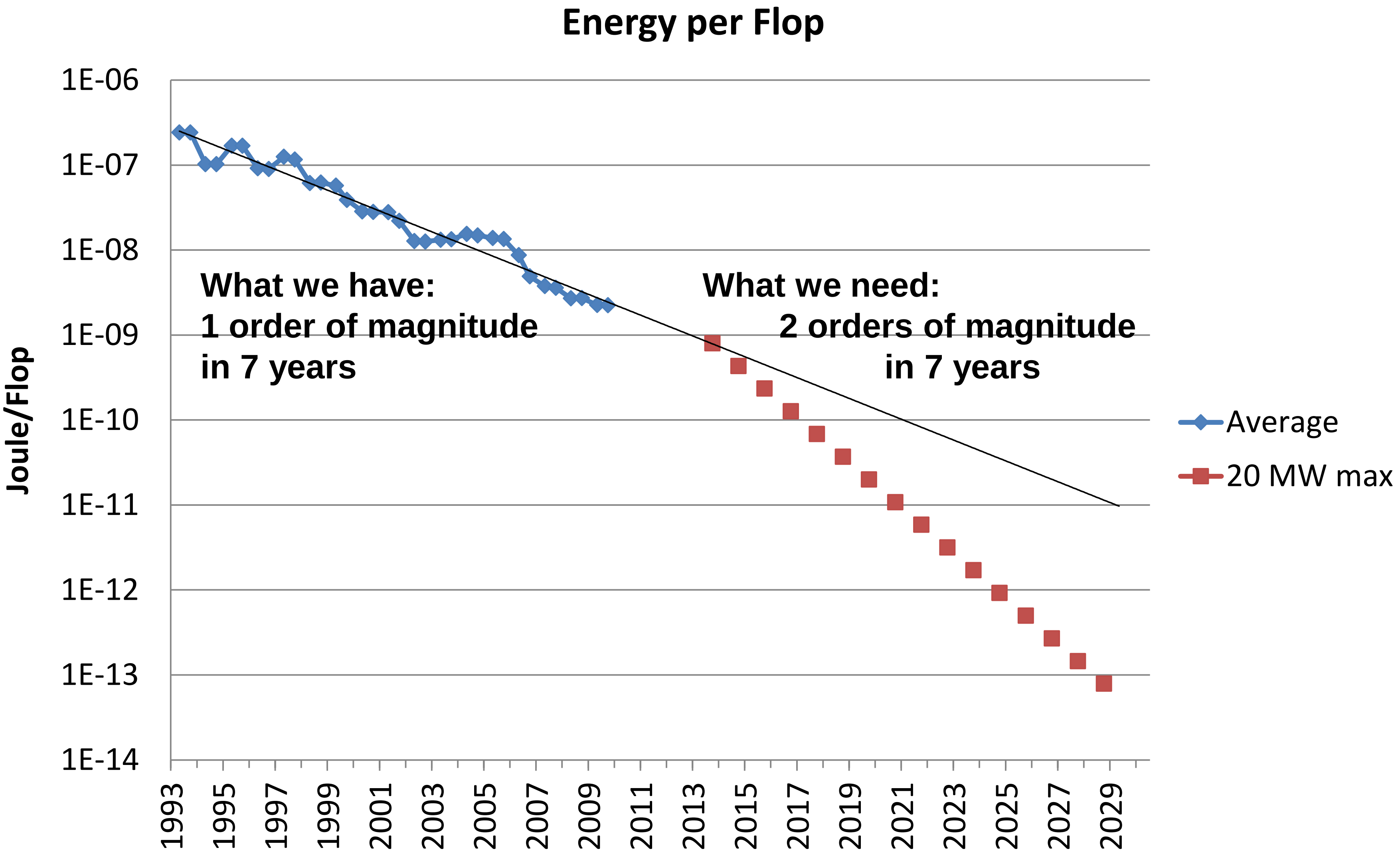
  → It's just really expensive!

[1] DOE, The Opportunities and Challenges of Exascale Computing, Summary Report of the ASCAC subcommitte, 2010.
[2] PNNL, Tackling the Power and Energy Wall for Future HPC Systems, 2013. http://www.hpcwire.com/2013/12/17/tackling-power-energy-wall-future-hpc-systems/

Extrapolation of Energy Consumption



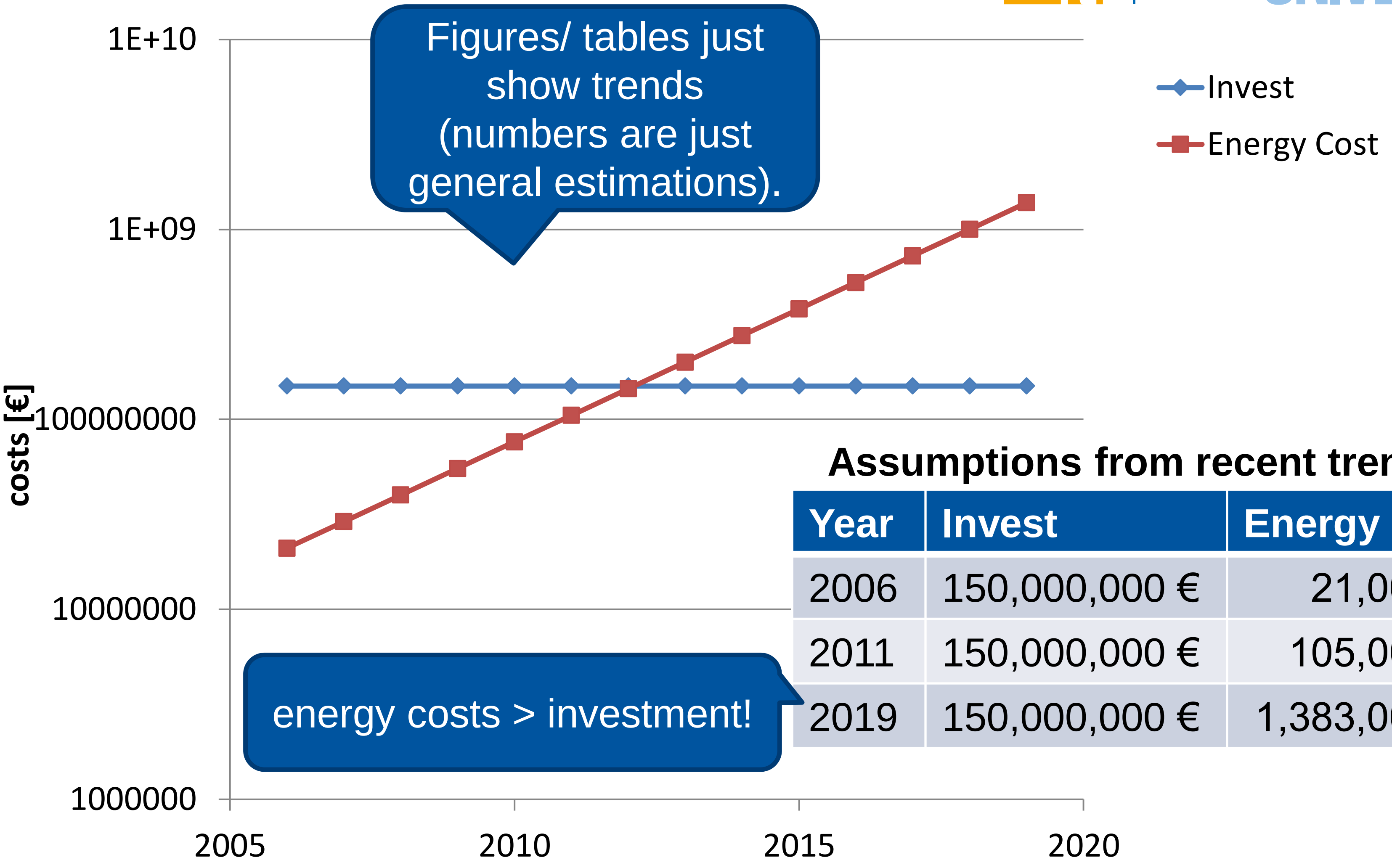
Energy Consumption Scenario: Max Power 20 MW



What we have:
1 order of magnitude
in 7 years

What we need:
2 orders of magnitude
in 7 years

Investment and power costs during 5 years



Figures/ tables just show trends (numbers are just general estimations).

energy costs > investment!

Assumptions from recent trends

Year	Invest	Energy
2006	150,000,000 €	21,000,000 €
2011	150,000,000 €	105,000,000 €
2019	150,000,000 €	1,383,000,000 €

Consequences of growing energy consumption

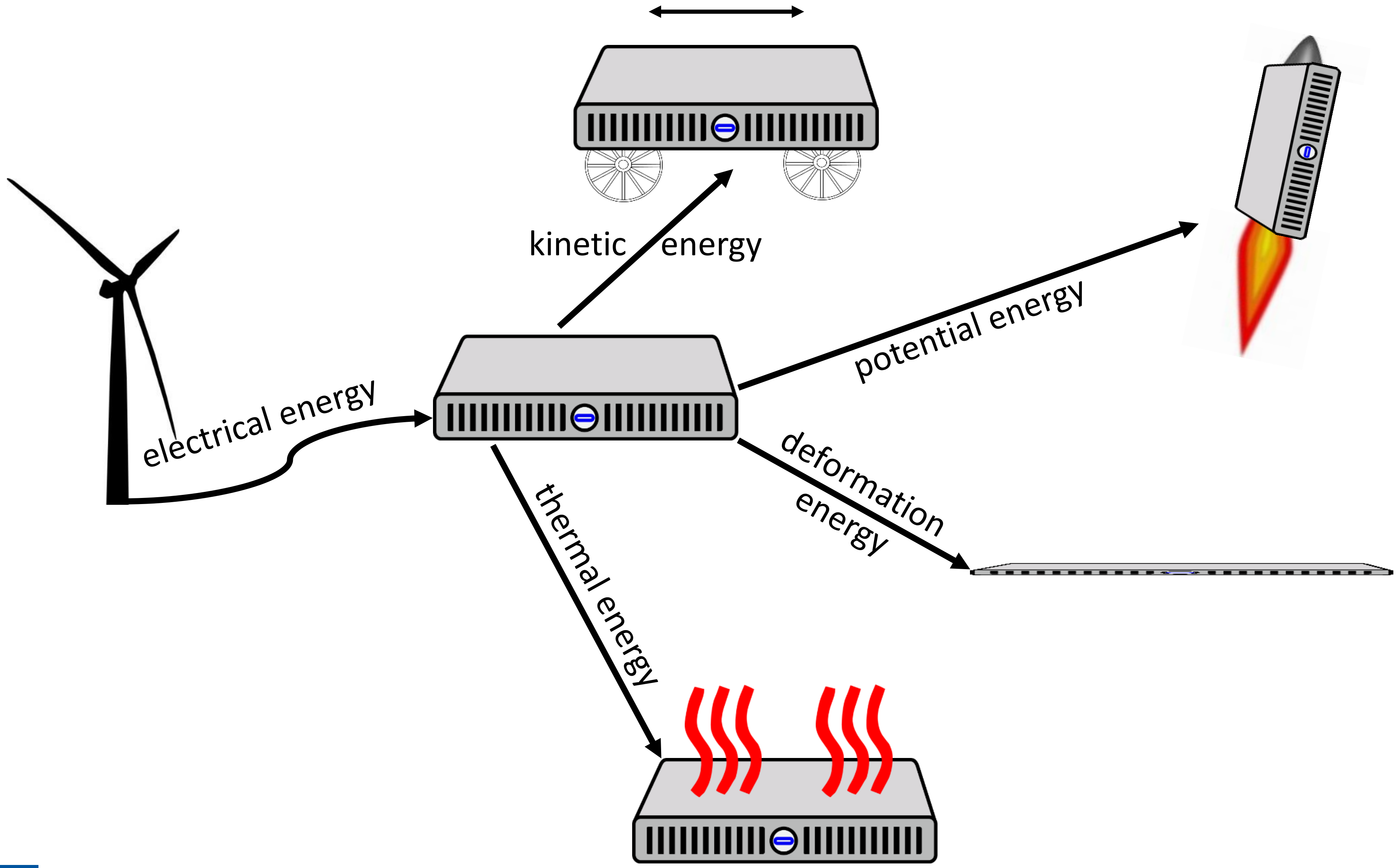
- The investment costs will become negligible
- Only a few centers will be able to operate and finance a supercomputer
- We will buy supercomputers like mobile phones:
  no contract with IBM but with energy supplier
- Performance optimization continues to be important, but additional energy efficiency constraints

Outline

1. Why supercomputers?
2. Modern processors
3. Basic optimization techniques for serial code
4. Data access optimization
5. Parallel computers
6. Parallelization and optimization strategies
7. Shared-memory programming with OpenMP
8. Distributed-memory programming with MPI
9. Hybrid programming (MPI + OpenMP)
10. Parallel algorithms
11. Heterogeneous architectures (GPUs, Xeon Phis)

12. **Energy efficiency**
    - → Motivation
        - → Towards recent HPC systems
        - → Trends
    - → **A (very short) side trip into physics**
    - → Energy efficiency of microarchitectures
    - → Benchmarking
    - → Outlook

Where does the electrical energy go?



Power Consumption

- **Power consumption P [W]:** $\boldsymbol{P \sim V^2 f}$
  - → Conversion of electricity to heat
  - → Frequency is dependent on voltage
  - → Often also: $\boldsymbol{P \sim f^3}$

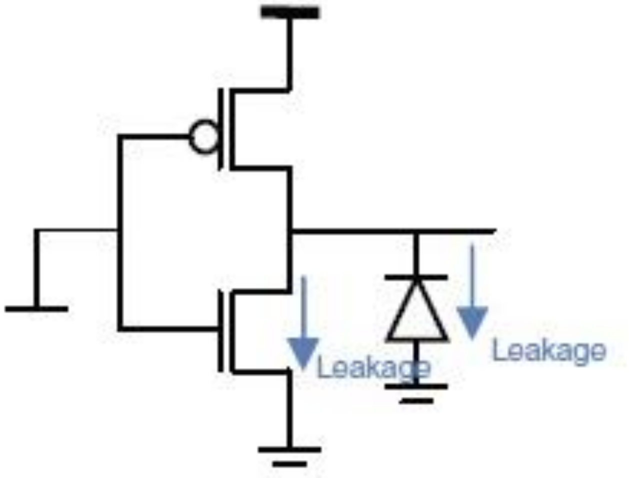
P: power consumption [W]
V: voltage [V]
f: clock frequency [Hz]

- **Two sources: static power $\boldsymbol{P_s}$ & dynamic power $\boldsymbol{P_d}$**
  - → $\boldsymbol{P = P_s + P_d}$

- **Impact on the infrastructure requirements:**
  - → design of power supplies and power distribution
  - → cooling infrastructure

Static Power Consumption P_s

- **Leakage of diodes and transistors**
  - → Increases due to the shrinking feature sizes of modern semiconductors
  - → Increases with temperature of transistors
  - → Trade-off: higher threshold voltage → higher switching speed (linear) → higher leakage (exponential)

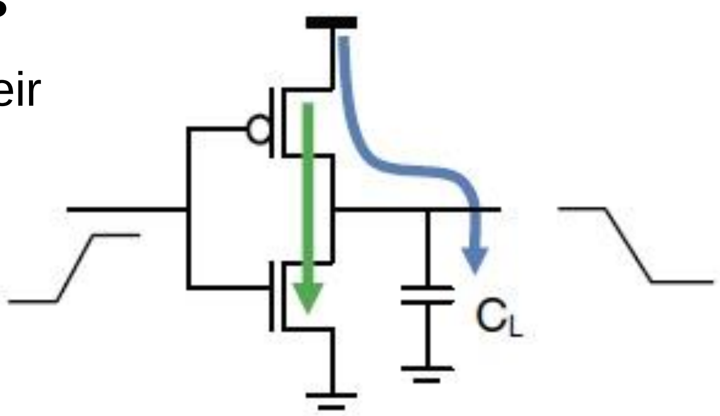


- **Ways to reduce P_s:**
  - → Reduce voltage supply: limited by technology (distinction of 1 & 0)
  - → Disable transistors: disable power to inactive components (power gating)
  - → Reduce number of transistors: opposite trend
  - → Improve transistor design & materials

Circuit diagram courtesy of Semiconductor Engineering

Dynamic Power Consumption P_d

- **Charging and discharging of capacitors**
  - → Depends on number of active cores and their switching activity



- **Ways to reduce P_d:**
  - → Reduce voltage supply: biggest impact
  - → Reduce frequency: reduces average power but not energy consumption
  - → Reduce switching activity/ cycles: data dependent, disable clocking of inactive components (clock gating)
  - → Reduce capacitance: approx. proportional to die size

Circuit diagram courtesy of
Semiconductor Engineering

Power vs. Performance Trade-off

- **Voltage reduction:**
  - → Decreases *P* quadratically (f const.)
  - → But: reduces transistor speed → latency is increased, limits max. frequency
  - → Dynamic voltage scaling: high V for performance critical path

- **Frequency reduction:**
  - → Decreases *P linearly* (f ~ V)
  - → But: increases execution time → const. *E*
  - → Dynamic frequency scaling: finding lowest frequency to finish task before deadline, voltage is decreased to minimum necessary

Energy Consumption

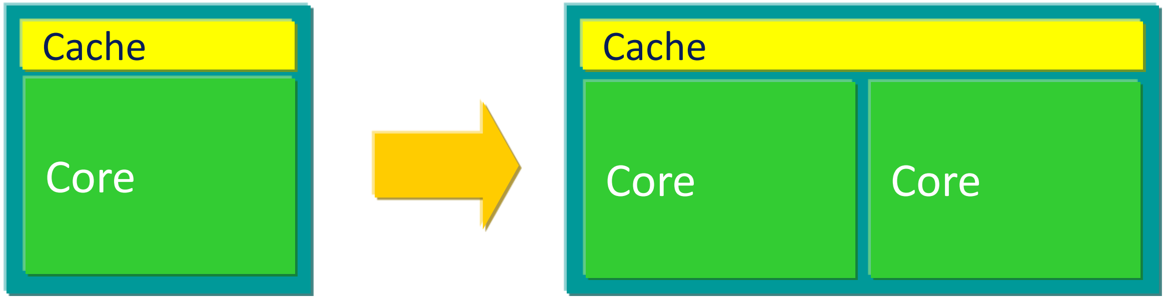
- **Electrical energy E [J]:** $\boldsymbol{E = \int P(t)dt}$
  - → Energy = power consumption * runtime
  - → Electricity cost, prize per kilowatt hour [€/kWh]

E: energy [J = W·s]
$P(t)$: power [W]
f: clock frequency [Hz]

- **Energy efficiency = energy / Flop [Joule]**
  - → Or: Mflops/Watt (= MFlop / energy)

Energy consumption – towards the multicore chip

- Can trade off frequency for parallelism
- Rule of thumb: Reduction of 1% voltage and 1% frequency reduces the power consumption by 3% and the performance by 0.66%.



Voltage	=	1	Voltage	=	-15%
Freq	=	1	Freq	=	-15%
Area	=	1	Area	=	2
Power	=	1	Power	=	~1
Perf	=	1	Perf	=	~1.8

(Based on slides from Shekhar Borkar, Intel Corp.)

Energy consumption – towards the multicore chip

f_c:	single core clock freq.
E:	its energy consumption
p:	its performance
$\varepsilon_f = \Delta f_c / f_c$:	rel. change in clock freq.
m:	# of ’slow’ cores
p_m:	multi-core performance
$\varepsilon_p = \Delta p / p$:	rel. change in perf.

- **Energy consumption $E \sim f_c{}^3$**
  - → Relative change in clockrate ε_f results in power dissipation of

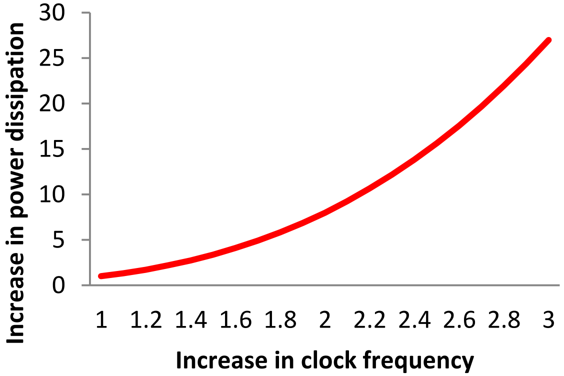
$$E + \Delta E = (1 + \varepsilon_f)^3 \mathrm{E}$$

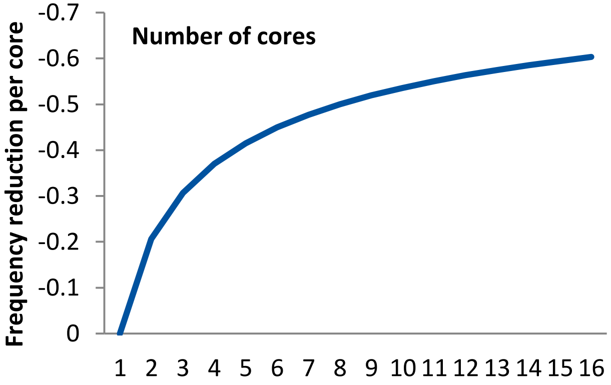
  - → Reducing the clock frequency opens the possibility of placing more than one CPU on the same die while keeping the same *power envelope* as before:

$$(1 + \varepsilon_f)^3 m = 1 \Leftrightarrow \varepsilon_f = \frac{1}{\sqrt[3]{m}} - 1$$

- **Overall performance of a multicore chip**

$$p_m = (1 + \varepsilon_p) p \cdot m \qquad \text{with } \varepsilon_p > \frac{1}{m} - 1$$





Energy Efficiency of Architectures

Machine Type	MFlops/Watt	Total Power [kW]
Xeon E5-2698v4 20C 2.2GHz, NVIDIA Tesla P100	9462.1	349.5
Xeon E5-2690v3 12C 2.6GHz, NVIDIA Tesla P100	7553.5	1312
Xeon E5-2618Lv3 8C 2.3GHz, PEZY-SCnp	6673.8	150
Sunway SW26010 260C 1.45GHz	6051.3	15371
Intel Xeon Phi 7210 64C 1.3GHz	5806.3	77
Intel Xeon Phi 7250 68C 1.4GHz	4985.7	2718.7
Intel Xeon Phi 7230 64C 1.3GHz	4688.0	1087
Intel Xeon E5-2680v2 10C 2.8GHz, Nvidia K80	4112.1	190
Intel Xeon Phi 7250 68C 1.4GHz	4086.8	748.1
Intel Xeon Phi 7230 64C 1.3GHz	3836.6	111

Source:Green500 11/2016

Energy consumption of principle operations

Property	Value
Energy/Flop on BG/Q	5.9 * 10^-10 J
Energy/Flop of Multiply Unit of Nehalem	2.2* 10^-10 J
Transfer of 1 Bit (IB, Mare Nostrum, Photonics)	5*10^-11 J
On board photonic transfer	10^-11 J
On chip photonic transfer	10^-12 J
On chip electric transfer via logic (130nm, 1.2V, ITRS2007)	0.1-1*10^-12 J
On chip electric transfer via interconnect (130 nm, 1.2V, ITRS2007)	3 * 10^-12 J
Off chip electric transfer via interconnect (130 nm, 1.2V, ITRS2007)	40-60*10^-12 J

Principal Limits

- **See work by**
  - → Lecture by John von Neumann (1949)
  - → R. Landauer, IBM J. Res. Develop, **3**, 183-191 (1961) „Irreversibility and Heat Generation in the Computing Process“
  - → Charles H. Bennet, Int. J. of Theoretical Physics, Vol. 21, No. 12, (1983) „The Thermodynamics of Computation – a Review“
  - → Richard P. Feynman, Nishina Memorial Lecture (1985) „The Computing Machines in the Future“
- **Landauer’s principle: changing one bit of information needs at least an amount of energy of $\boldsymbol{ln(2) \cdot k \cdot T}$**
- $\boldsymbol{ln(2) \cdot k \cdot T}$ **per Bit means at 300K:**
  - → Limit of: 2.870981796E−21 J/Bit
  - → Limit of: 1.837428349E−19 J/Flop
  - → Remember: Green Top 1 system is at: 1.90E-10 J/Flop

k: Boltzmann constant
T: absolute temperature (in Kelvin)
K: Kelvin (300K ~ room temperature)

Outline

1. Why supercomputers?
2. Modern processors
3. Basic optimization techniques for serial code
4. Data access optimization
5. Parallel computers
6. Parallelization and optimization strategies
7. Shared-memory programming with OpenMP
8. Distributed-memory programming with MPI
9. Hybrid programming (MPI + OpenMP)
10. Parallel algorithms
11. Heterogeneous architectures (GPUs, Xeon Phis)

12. **Energy efficiency**
    - → Motivation
        - → Towards recent HPC systems
        - → Trends
    - → A (very short) side trip into physics
    - → **Energy efficiency of microarchitectures**
    - → Benchmarking
    - → Outlook

Parameters that Affect Energy Efficiency

- **There is a big parameter space**
  - → Number of cores
  - → Clock frequency (of each core)
  - → Prefetcher (enable/disable, various configurations)
  - → Symmetric multi-threading
  - → Branch prediction
  - → GPU clock frequency
  - → Cache size
  - → Speed of (on-chip) interconnects
- **We need to find sweet spots**
  - → Not only for each application, but for each application phase
  - → Finding sweet spots may not consume much energy!

Power management of microarchitectures

P: power consumption
V: voltage
f: clock frequency

- **Recap:** $P \sim V^2 f$
- **Aim: reduce power consumption**

- **Microarchitectures provide different power states**
  - → C-states: CPU power state
  - → P-states: CPU performance state
  - → ….
- **View/ control from different levels**
  - → Core level
  - → Processor level
  - → OS level
  - → Application level

More: http://software.intel.com/en-us/articles/list-of-useful-power-and-power-management-articles-blogs-and-references

C-states

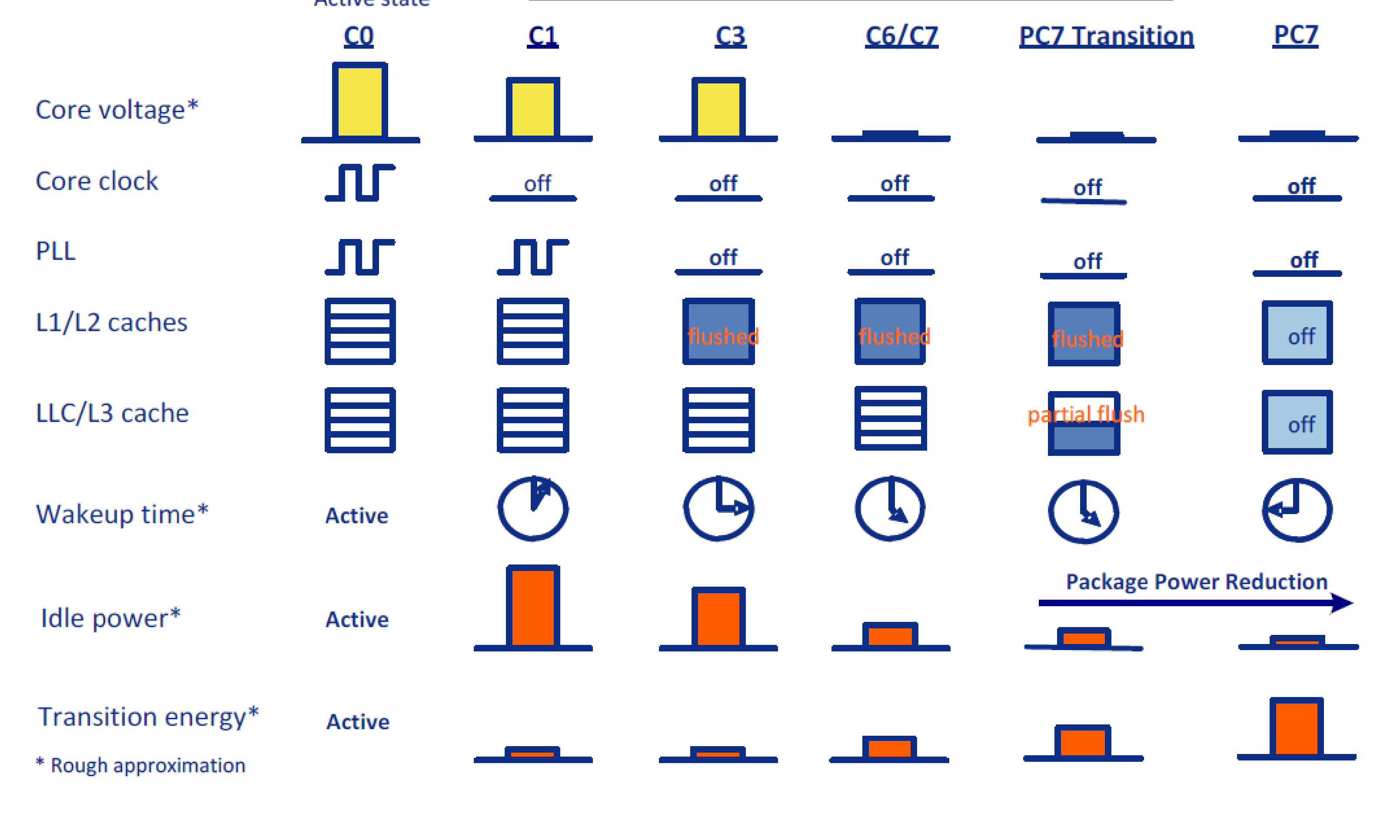
- **C0 (operating state)**
  - → CPU is active and busy doing something
- **If not doing anything useful, shut it down → idle states**
- **C1 (idle state: halt)**
  - → Clock is prevented from reaching the core
- **C2 (idle state: stop-clock)**
  - → External I/O controller hub blocks interrupts to the processor
- **C3, C4… (idle state: deep sleep)**

- **Core C-state: CC1,…**
  - → Own state per core
- **Processor/ package C-state: PC1,…**
  - → Across all cores of package

→ **C-states disable certain transistors power**

Example C-states (Intel)

C0: active state executing code
C1: halted, snoops serviced
C3: Core (L1/L2) caches flushed
C6: Core state saved and Core voltage reduced to ~0
C7: When last core enters C7, LLC is flushed progressively
PC0: active package level state
PC1: low latency package level state
PC7 (Depper Power down): LLC fully shrunk, no snoops



Source: Intel, Energy-Efficient Platforms – Considerations for Application Software and Services, 2011. http://software.intel.com/sites/default/files/94/31/38273

P-states

- **E.g. laptop in low power profile and operating on battery**

- **All P-states are operational states**
  - → C-state C0, core/processor can do useful work in any P-state
- **P-States: switch between supported frequencies & voltages**
  - → Higher P-state → lower C0 operating frequency/ voltage
    - →Reduce speed of processor
    - →Application will run longer
- **Reduced power consumption, reduced performance**

$$P \sim V^2 f$$

P: power consumption
V: voltage
f: clock frequency

- ***Dynamic Voltage and Frequency Scaling (DVFS)***

Example P-States

AMD Istanbul CPU

P-State	Multiplier	Voltage	Frequency	*f*	V^2*f
P-State 0:	13.00000	1225 mV	2600 MHz	3.25	4.42
P-State 1:	10.50000	1175 mV	2100 MHz	2.63	3.30
P-State 2:	8.50000	1150 mV	1700 MHz	2.13	2.56
P-State 3:	7.00000	1125 mV	1400 MHz	1.75	2.01
P-State 4:	4.00000	1050 mV	800 MHz	1.00	1.00

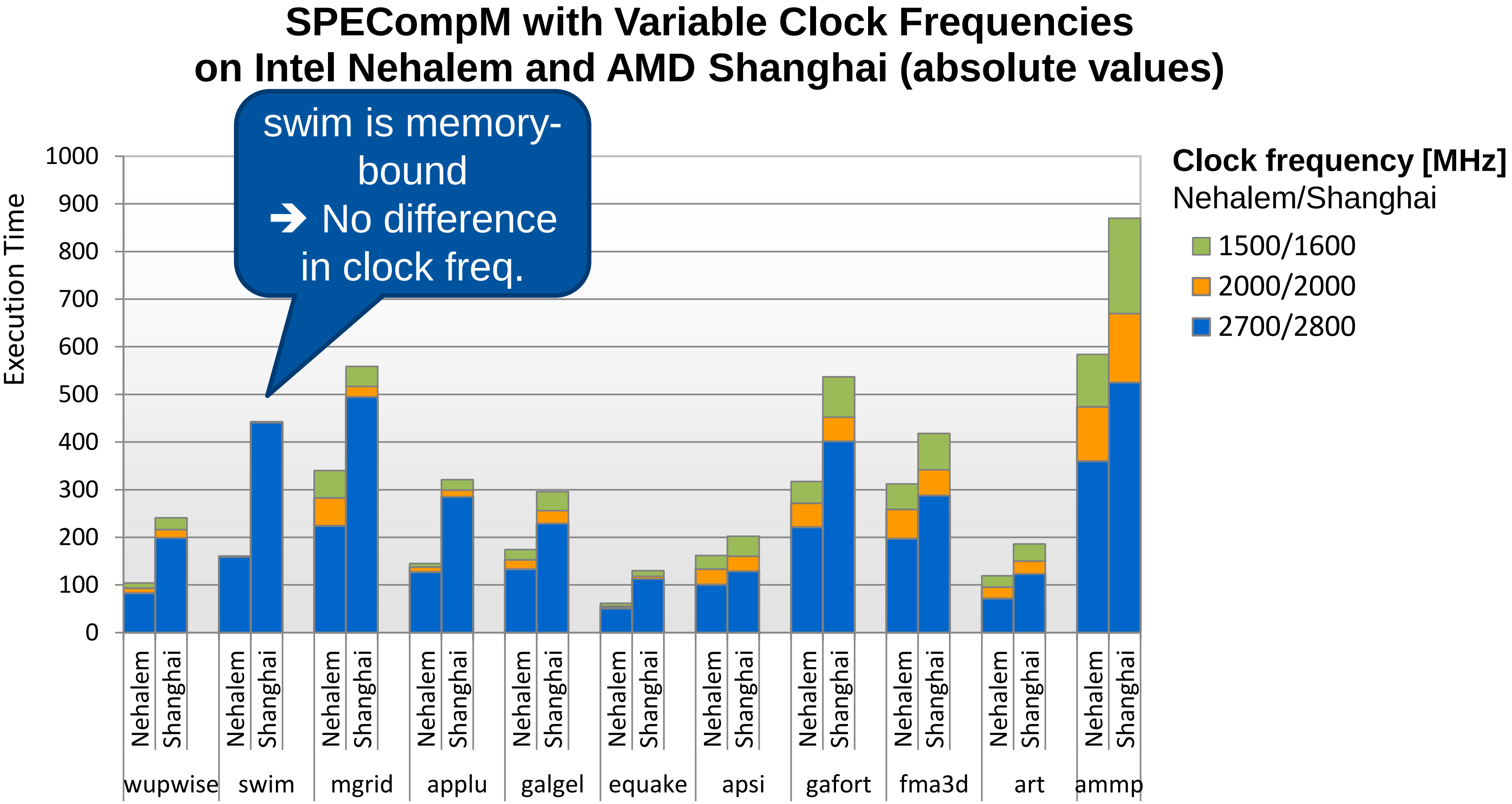
4 times more power consumption needed in P0 than in P4.

Intel Pentium M 1.6 GHz

P-State	Multiplier	Voltage	Frequency	*f*	V^2*f	Power [W]
P-State 0:	8.00000	1484 mV	1600 MHz	2.66	11.04	25
P-State 1:	7.00000	1420 mV	1400 MHz	2.33	8.09	~17
P-State 2:	6.00000	1276 mV	1200 MHz	2.00	5.34	~13
P-State 3:	5.00000	1164 mV	1000 MHz	1.66	3.38	~10
P-State 4:	4.00000	1036 mV	800 MHz	1.33	1.93	~8
P-State 5:	3.00000	956 mV	600 MHz	1.00	1.00	6

Source: http://download.intel.com/design/network/papers/30117401.pdf

Runtime-impact of Various Clock Frequencies



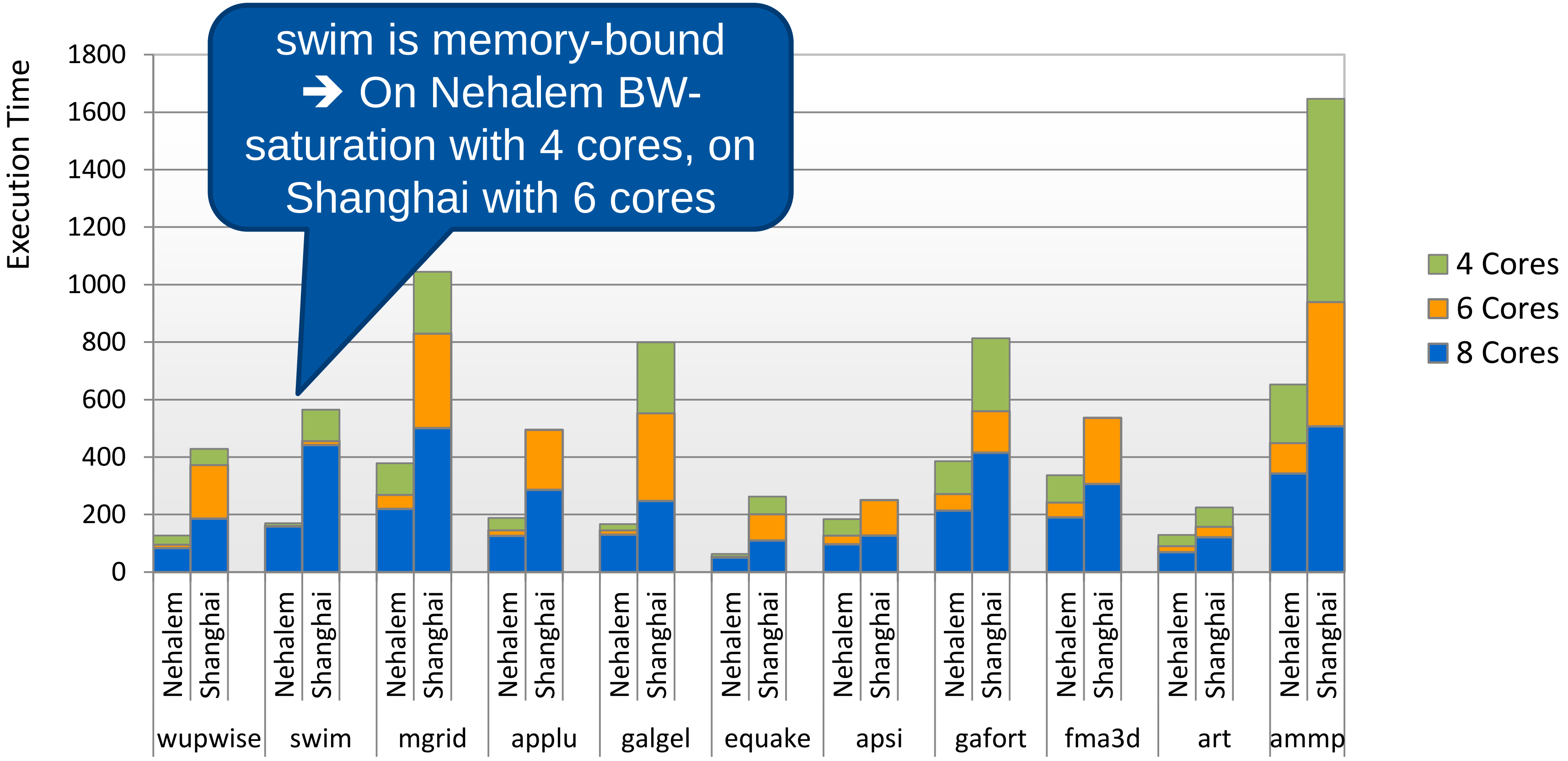
swim is memory-bound
➔ No difference in clock freq.

Further reading: D. Hackenberg, D. Molka and W. E. Nagel, Comparing Cache Architectures and Coherency Protocols on x86-64 Multicore SMP Systems, In Proceedings of the 42nd International Symposium on Microarchitecture (MICRO'09), IEEE/ACM, 2009

Runtime-impact of Various Core Counts

Lecture *Performance and correctness analysis of parallel programs* will cover the **Roofline** model to assess whether application are memory or compute bound.

SPECompM with Variable Core Counts on Intel Nehalem and AMD Shanghai (absolute values)



swim is memory-bound
➔ On Nehalem BW-saturation with 4 cores, on Shanghai with 6 cores

Further reading: D. Hackenberg, D. Molka and W. E. Nagel, Comparing Cache Architectures and Coherency Protocols on x86-64 Multicore SMP Systems, In Proceedings of the 42nd International Symposium on Microarchitecture (MICRO'09), IEEE/ACM, 2009

Outline

1. Why supercomputers?
2. Modern processors
3. Basic optimization techniques for serial code
4. Data access optimization
5. Parallel computers
6. Parallelization and optimization strategies
7. Shared-memory programming with OpenMP
8. Distributed-memory programming with MPI
9. Hybrid programming (MPI + OpenMP)
10. Parallel algorithms
11. Heterogeneous architectures (GPUs, Xeon Phis)

12. **Energy efficiency**
    - → Motivation
        - → Towards recent HPC systems
        - → Trends
    - → A (very short) side trip into physics
    - → Energy efficiency of microarchitectures
    - → **Benchmarking**
    - → Outlook

HPC Power Consumption Benchmarks

- **The HPC community needs power consumption benchmarks!**

- **How about the Green500 list?**
  - → Ranks all TOP500 systems
  - → Metric: MFLOPS/Watt (higher is better)

What we have: The Green500 List

- **1.7 Subcomponent:** A subcomponent is that part of a supercomputer which can be measured in isolation for power consumption. For example, this may be a single chassis, a rack, or any physical enclosure that is larger than the fundamental unit of a chassis. The important criterion here is the ability to measure power in isolation for a period of time while the subcomponent is engages collectively with the entire supercomputer to perform a task. Single subcomponent power consumption under load will be used to estimate system-wide power consumption. The entire system may be considered as a "subcomponent" under this definition. *The subcomponent that is measured must consume at least 1 kilowatt of power.*

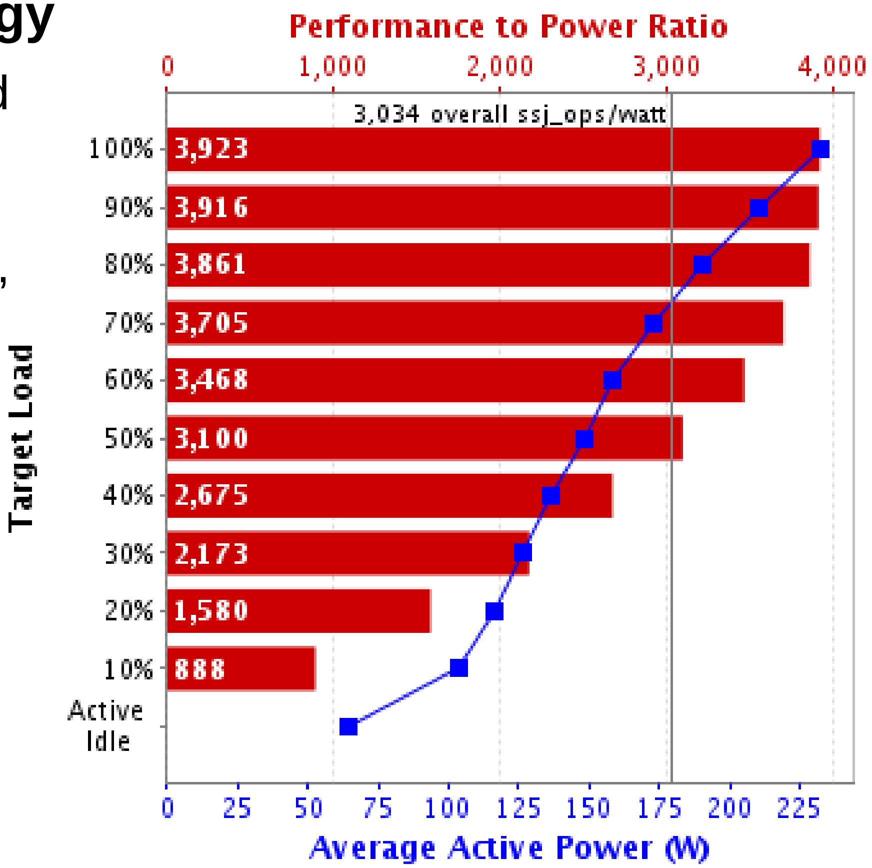
- **Power extrapolation: $P = N \cdot P_{subcomponent}$**
- **Assuming that**
  1. the computational workload during the Linpack benchmark is well-balanced across all units, and
  2. all units are identical and consume the same amount of power for the same workload.

Are these assumptions valid?

Source: http://www.green500.org/docs/pubs/RunRules_Ver1.0.pdf

What we have: SPECpower_ssj2008

- **Industry-standard benchmark**
  - → that evaluates the power and performance characteristics of volume server class and multi-node class computers
- **Very mature benchmark methodology**
  - → Dedicated power daemon for accepted power meters only
  - → Records current, voltage, power factor, temperature
  - → Very strict run rules
- **BUT**
  - → Java Workload
  - → Strongly VM dependent
  - → Servers typically run Windows



What we need

- **An HPC benchmark with a sophisticated power measurement methodology**

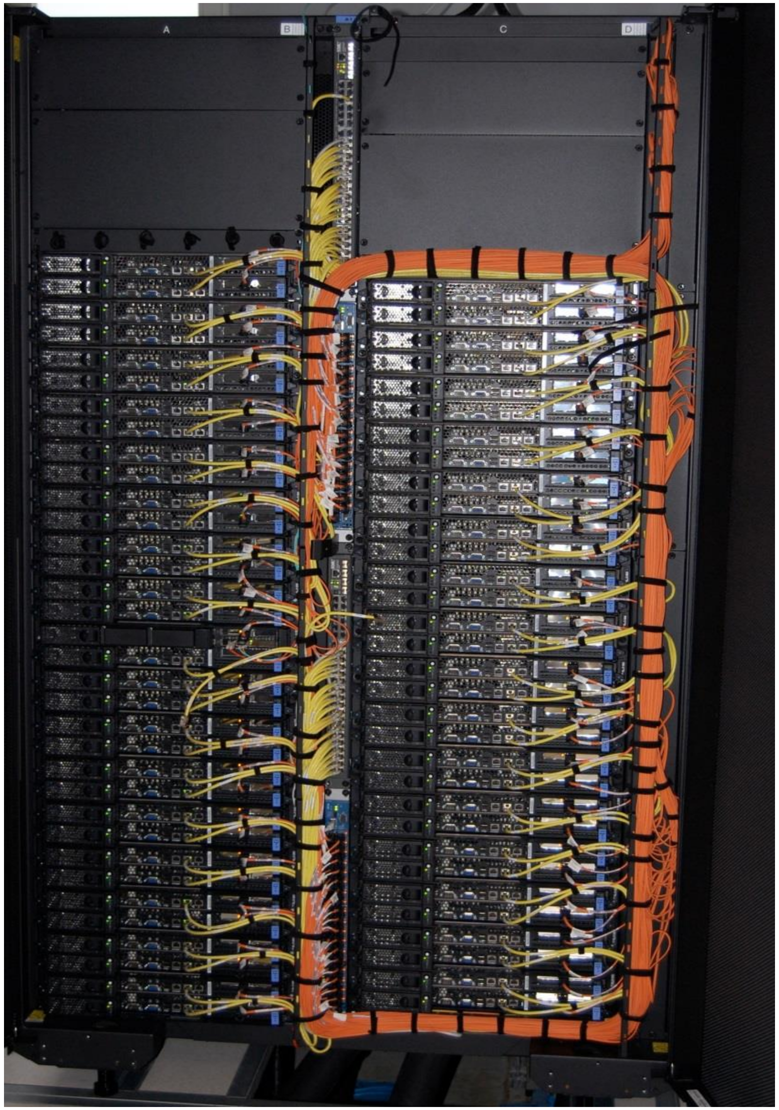
- **SPEC MPI2007**
  - → Industry-standard HPC benchmark
  - → Averages the results of 12 MPI applications that are common in HPC
  - → Medium data set scales up to 128 MPI ranks, runs up to 512 ranks
  - → Large data set scales up to 2048 ranks, tested up to 4096 ranks

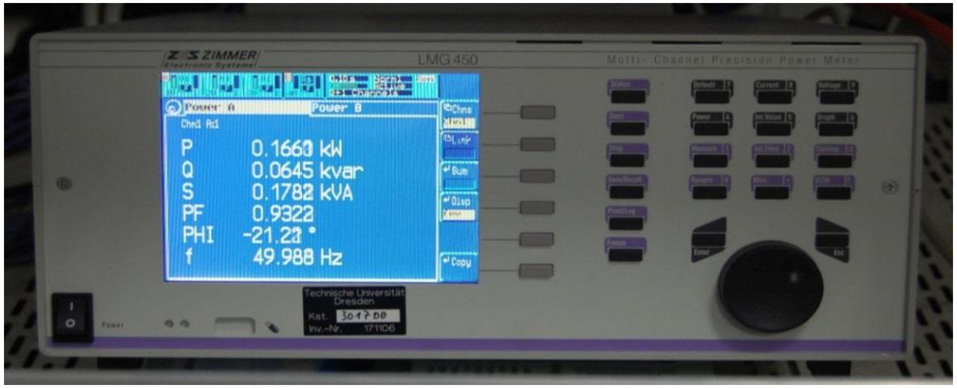
- **Challenge**
  - → Companies probably do not want to measure each node
  - → We need run rules that ensure that hardware vendors do not cheat (too much)

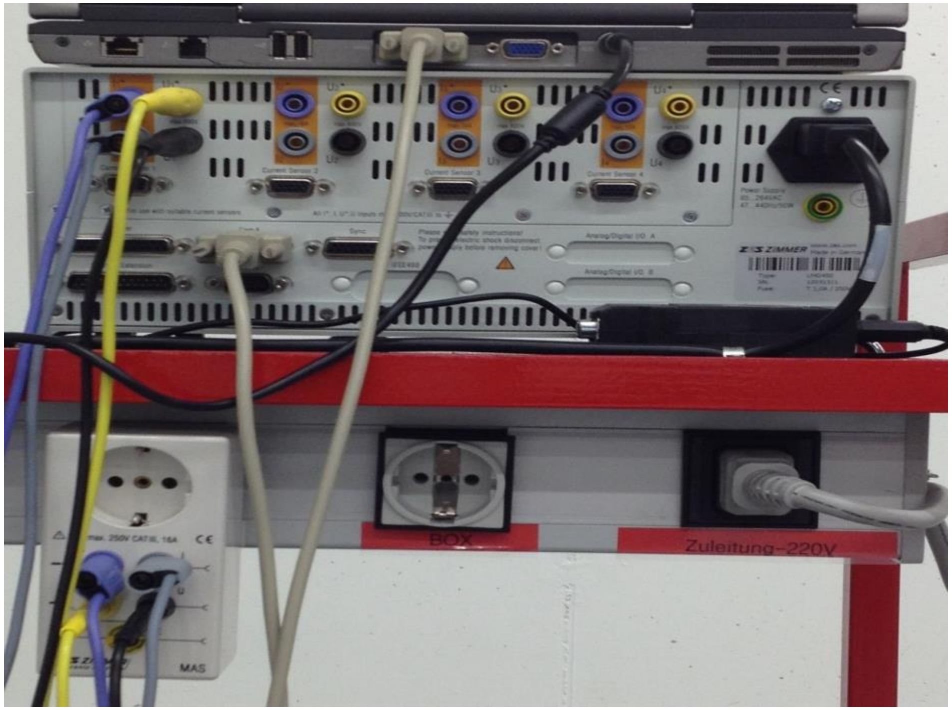
Test Equipment Overview

- **Test System: IBM iDataPlex**
  - → 32 nodes, each with 12x 4 GB DDR3, 250 GB HDD, QDR IB
  - → 64x Intel Xeon E5530, 2.4 GHz, 80W TDP
  - → Turbo Boost up to 2.66 GHz, HyperThreading off
- **Power Meter**
  - → 1x ZES LMG 450 (4 channel)
  - → 1x ZES LMG 95 (1 channel)
  - → Measuring compute nodes only, no switches, no I/O
- **Benchmark: SPEC MPI2007 V2.0**
  - → Medium data set
  - → Intel Compiler Suite 11.1, Open MPI 1.4.1

Test System & Power Measurements




Z·S ZIMMER Electronic Systems
LMG 450
Multi-Channel Precision Power Meter
Power A
Power B
P 0.1660 kW
Q 0.0645 kvar
S 0.1782 kVA
PF 0.9322
PHI -21.22 °
f 49.988 Hz
Technische Universität Dresden



Test System & Power Measurements