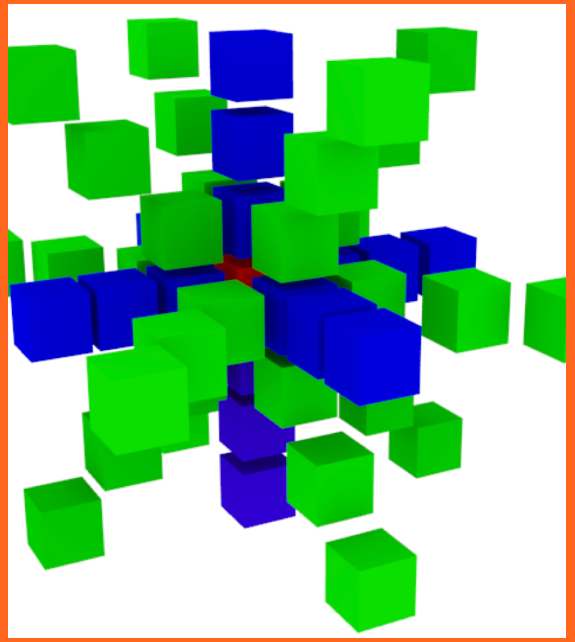
CS-E4690 – Programming Parallel Supercomputers

Current HPC landscape

Maarit Korpi-Lagg

maarit.korpi-lagg@aalto.fi

Aalto University
School of Science



Why HPC & What is it?

Task-oriented definition (currently)

> ***"A computer system designed to execute applications that would take days/years/centuries on a desktop/mobile device, in seconds/minutes or require weeks or months to run, even at large scale."***

Why HPC & What is it?

Hardware-oriented definition (currently)

> ***Network of processing elements (PEs, currently of 10k or more) that enable them to process the task in parallel. In a common use case, the PEs need to exchange data, hence the HPC system needs to have fast memories and low-latency, high-bandwidth communication systems between the PEs and PEs and the associated memories (>100Gb/s).***

Parallelism==computation + communication and making them concurrent

Example

- Let us assume that you want to solve a dense linear system of the form $Ax = b$, where A is a $n \times n$ matrix, and b is a n vector.

- You plan to use an unoptimized algorithm with computational complexity of $\mathcal{O}(n^3)$.

- You have a PC that is capable of computing 50 billion floating point multiplications a second.

How long will this take?

n	required ops	time
10	1000	20 ns
100	1M (10^6)	20 μs
1,000	1G (10^9)	20 ms
10,000	1T (10^{12})	20s
100,000	1P (10^{15})	20,000s=5,5h
1,000,000	1E (10^{18})	20,000,000s=231d

If you have such processors working in parallel, the large-scale problem becomes computable

How long will this take?

If you have such processors working in parallel, the large-scale problem becomes computable *); ideally, doubling the number of processors will halve the computing time

n	nproc	time
1,000,000	1	231d
-”-	2	116d
-”-	4	58d
…	…	…
-”-	1024	5,4h

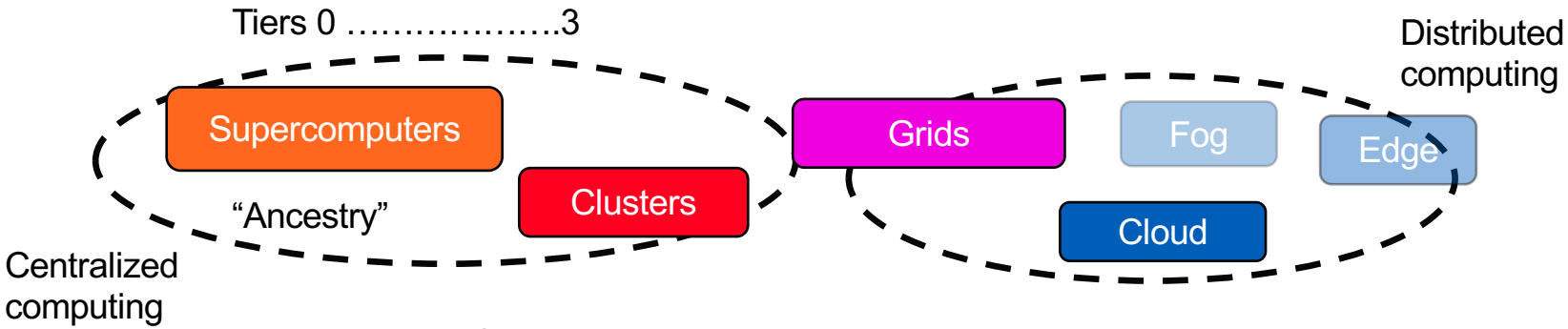
***) in this case, however, you should also optimize your algorithm, too.**

Why HPC & What is it?

- (IMO) HPC==any computation or data analysis task that you cannot perform in a standard desktop computer at hand
- This course: we learn to use supercomputing environments *)
  - **Minimalistic** computing environment
  - Constant, **rapid changes** to the environment – code becomes obsolete if you do not maintain it
  - Parallel programming is **hard** – producing scalable and optimized code is an effort
  - HPC **paradigms change fast** – need to be ready for complete turnaround (currently from homogeneous to heterogeneous systems)

*) HPC in cloud/edge/fog computing is another story, used in big data science; you learn about that, e.g., in course CS-E4640

A rough sketch of the current HPC landscape



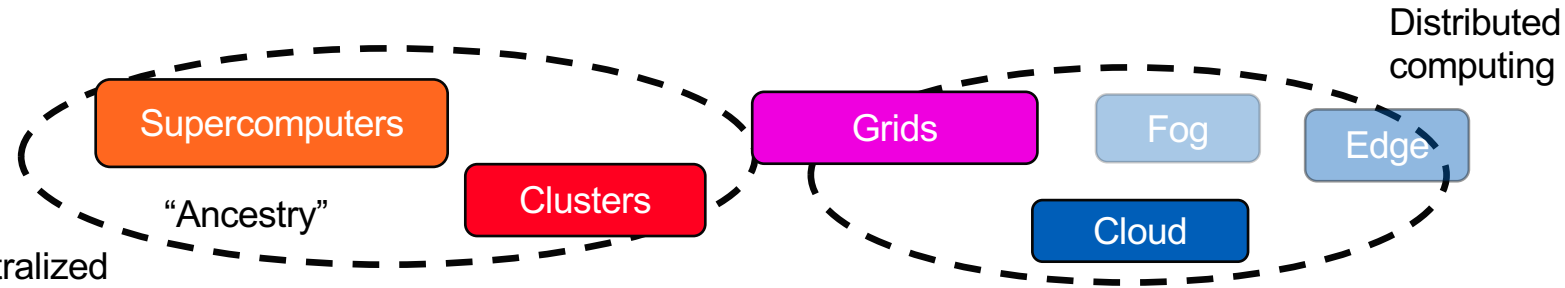
Coupled
Centralized resource allocation and handling
Large, single jobs
Non- or partly virtualized
Data is difficult to access
Hard to use
Non-resilient
Non-scalable
Expensive
Research

VS

Loose
Decentralized
Small or medium
Fully virtualized
Data is easy to access
User friendly
Fault tolerant
Scalable
Affordable
Business

Sadashiv & Kumar (2011)

A rough sketch of the current HPC landscape



PRACE

Tier-0

LUMI

Mahti

Puhti

DECI

Tier-1

Triton

Tier-2

Tier-3

HTCondor@Aalto

FCCI

Euro-Grid

Rahti

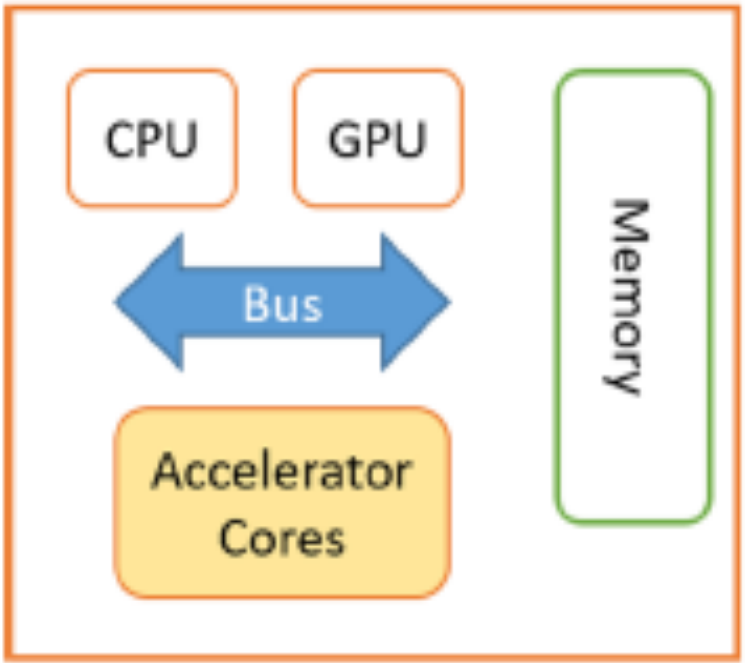
Cpouta

Microsoft Azure

Amazon AWS

Read more from the websited linked to in the core material

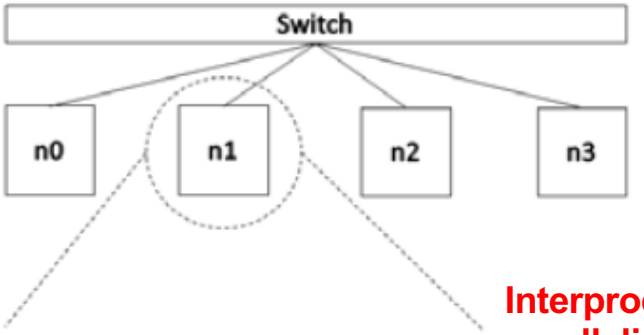
Current paradigm: Heterogeneity



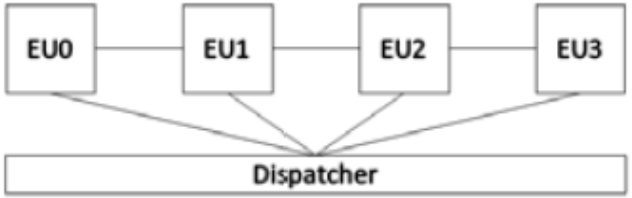
Heterogeneity can be predicted to increase with the inclusion of even more specialized hardware components with built-in logic for interfacing

Cardwell+(2020)

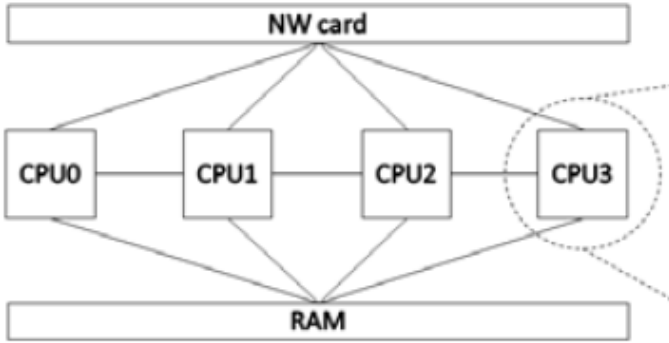
Current paradigm: NUMA



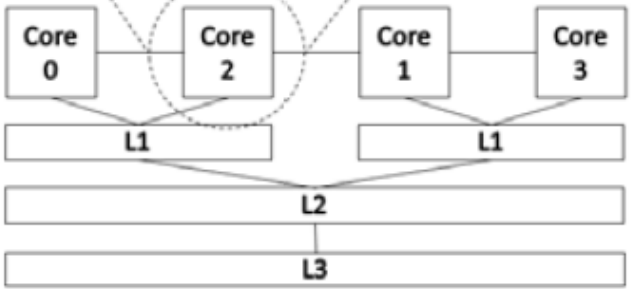
Non-Uniform Memory Access: Access to one's own memory is faster than somebody else's



Interprocessor parallelism



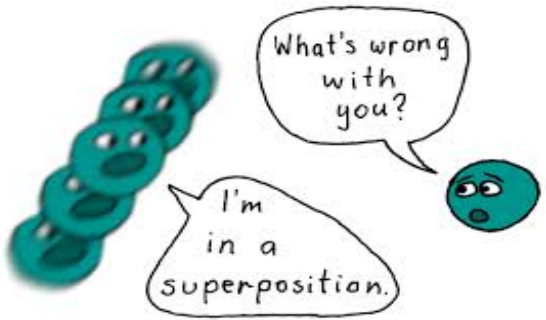
Intraprocessor parallelism



Supalov (2018), Chapter 1

Next paradigm: quantum computers

- From bits (transistor values of ‘1’ and ‘0’) to qubits (quantum states of, e.g., electron in superposition of ground (0) or excited states (1))
- Classical logic is replaced by the rules of quantum mechanics
- Quantum effects: Superposition – interference – entanglement
- Cubits are extremely sensitive – building and operating a stable quantum computer will not be trivial
- Currently < 100 cubits,



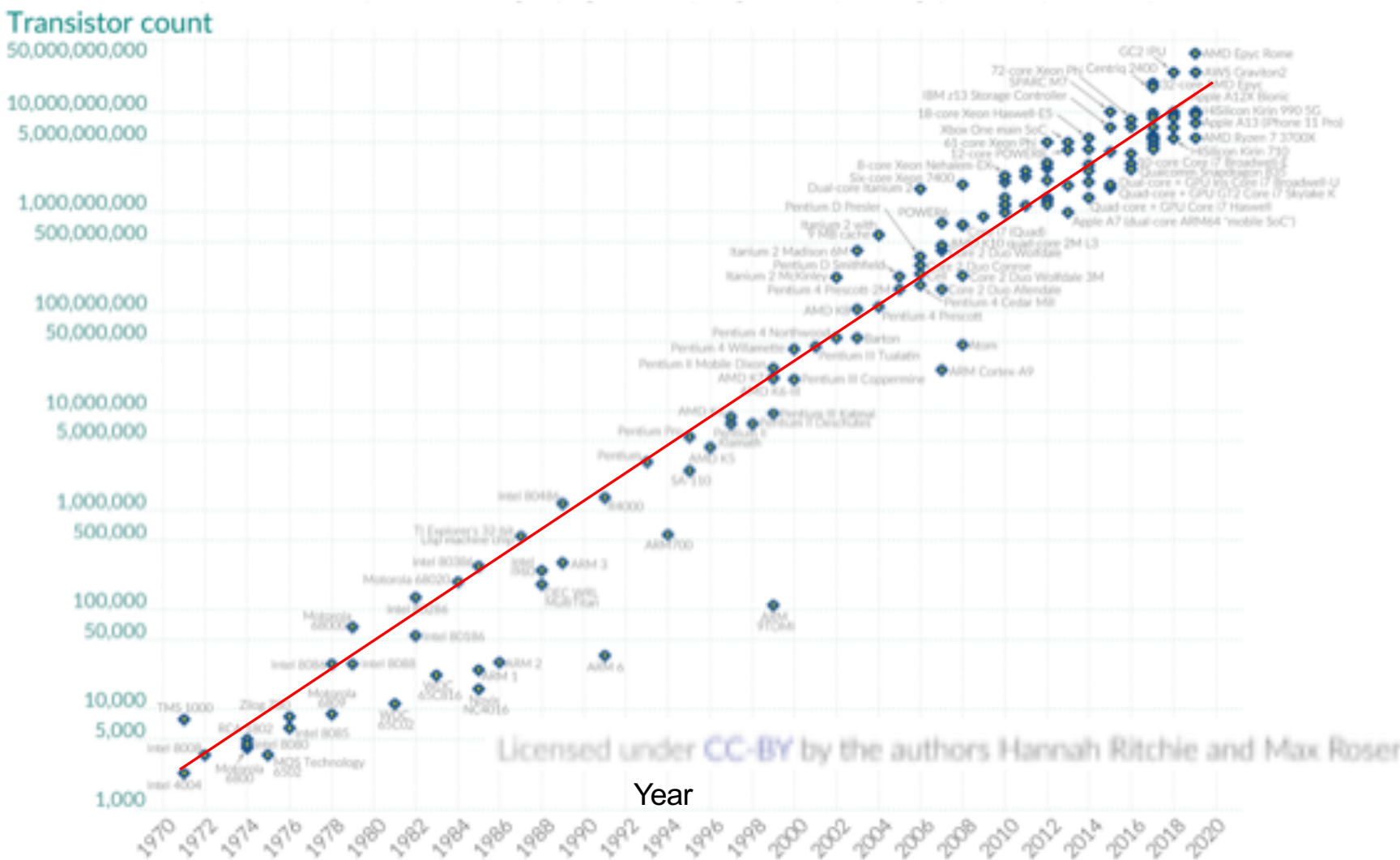
Credit: CC BY-SA 3.0

Quantum advantage – a few years
Quantum supremacy – decade

Moore's first law

"The number of transistors per chip doubles every two years"

Gordon Moore (1965)


Transistor count
50,000,000,000
10,000,000,000
5,000,000,000
1,000,000,000
500,000,000
100,000,000
50,000,000
10,000,000
5,000,000
1,000,000
500,000
100,000
50,000
10,000
5,000
1,000
Year
1970
1972
1974
1976
1978
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
2004
2006
2008
2010
2012
2014
2016
2018
2020
Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

A

Moore's second law

Although the cost for consumers has been constant….

The capital cost of a semiconductor fab has also increases exponentially over time

80 microns to few dozen nanometers
few million to 10 billion by 2018

Dennard scaling

Power density $\frac{P}{A}$ in CMOS transistors is constant to the first order when their size is reduced (Dennard, 1974)

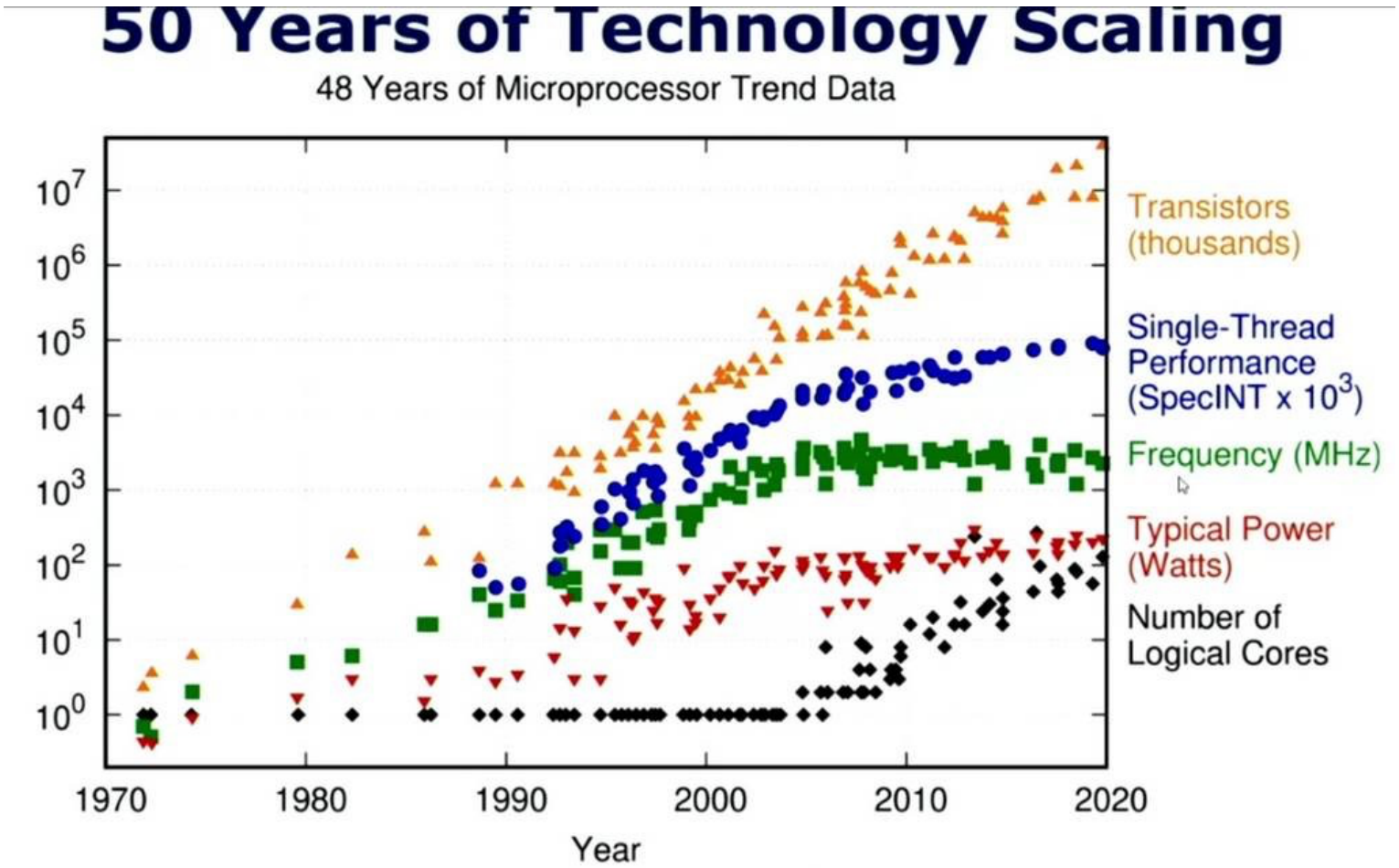
- $P \propto aCfU^2$; a is the switching frequency, C the capacitance, f the clock frequency, and U is the voltage.
- Making the transistors smaller, their power use remains in proportion with the area.
- The clock frequency can be increased when shrinking the size, keeping the power consumption roughly the same.
- Hence, Dennard's recipe to make ever faster chips was to **make the transistor smaller.**

Power wall

CMOS chip design evolution: Dennard scaling does no longer hold.

- $P \propto aCfU^2 + \boldsymbol{I_{leakage}U + I_{short\ circuit}U}$
- When the voltage is reduced, higher-order effects, namely power losses become important, and finally dominant
- These heat up the chip, and lead to further power losses.
- The transistor no longer operates reliably, that can finally lead to thermal runaway.

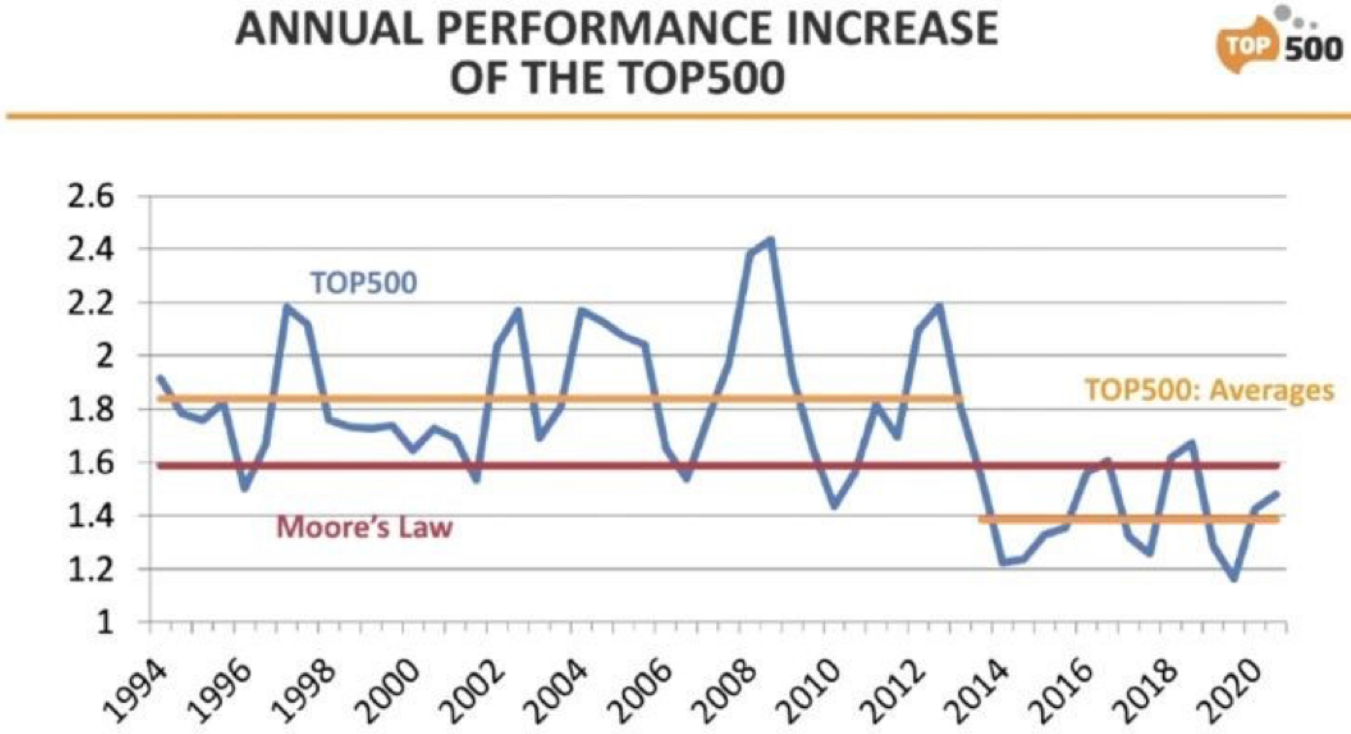
More in Bose (2011)



Aalto University
School of Science

Shekhar Borkar, Qualcomm's senior director of technology and a former Intel Fellow, Supercomputing 2020 conference

Trend seen in top500 list



Erich Strohmaier, senior scientist at Lawrence Berkeley National Laboratory

Materials (core)

N. Sadashiv and S. M. D. Kumar, "Cluster, grid and cloud computing: A detailed comparison," *2011 6th International Conference on Computer Science & Education (ICCSE)*, 2011, pp. 477-482, doi: 10.1109/ICCSE.2011.6028683, and references therein.

Aalto scientific computing: https://scicomp.aalto.fi/about/

CSC scientific computing: https://research.csc.fi/csc-s-servers

TOP500 supercomputer list https://www.top500.org/

PRACE resources: https://prace-ri.eu

FCCI infrastructure: https://www2.helsinki.fi/en/infrastructures/fcci

Materials (extra)

Asch, M. et al. (2018) ‘Big data and extreme-scale computing: Pathways to Convergence-Toward a shaping strategy for a future software and data ecosystem for scientific inquiry’, The International Journal of High Performance Computing Applications, 32(4), pp. 435– 479. doi: 10.1177/1094342018778123.

Guidi, G., Ellis, M., Buluç, A. Yelick, K., and Culler, D. 2021. 10 Years Later: Cloud Computing is Closing the Performance Gap. In Companion of the ACM/SPEC International Conference on Performance Engineering (ICPE '21). Association for Computing Machinery, New York, NY, USA, 41–48. DOI:https://doi.org/10.1145/3447545.3451183

Bose P. (2011) Power Wall. In: Padua D. (eds) Encyclopedia of Parallel Computing. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-09766-4_499

http://www.cs.virginia.edu/~robins/The_Limits_of_Quantum_Computers.pdf

https://www.csc.fi/fi/web/training/-/quantum-computing

CSC’s Quantum Learning Machine: https://research.csc.fi/-/kvasi

Materials (extra)

Cardwell, S.G. *et al.* (2020). Truly Heterogeneous HPC: Co-design to Achieve What Science Needs from HPC. In: Nichols, J., Verastegui, B., Maccabe, A.‘., Hernandez, O., Parete-Koon, S., Ahearn, T. (eds) Driving Scientific and Engineering Discoveries Through the Convergence of HPC, Big Data and AI. SMC 2020. Communications in Computer and Information Science, vol 1315. Springer, Cham. https://doi.org/10.1007/978-3-030-63393-6_23; also the whole conference proceedings book is an interesting read.

Supalov, A., 201, Inside the Message Passing Interface: Creating fast communication libraries, Walter de Gruyter Inc., Boston/Berlin https://doi.org/10.1515/9781501506871; recommendation for wannabe MPI geeks