NHR@FAU HPC Café
September 17, 2024

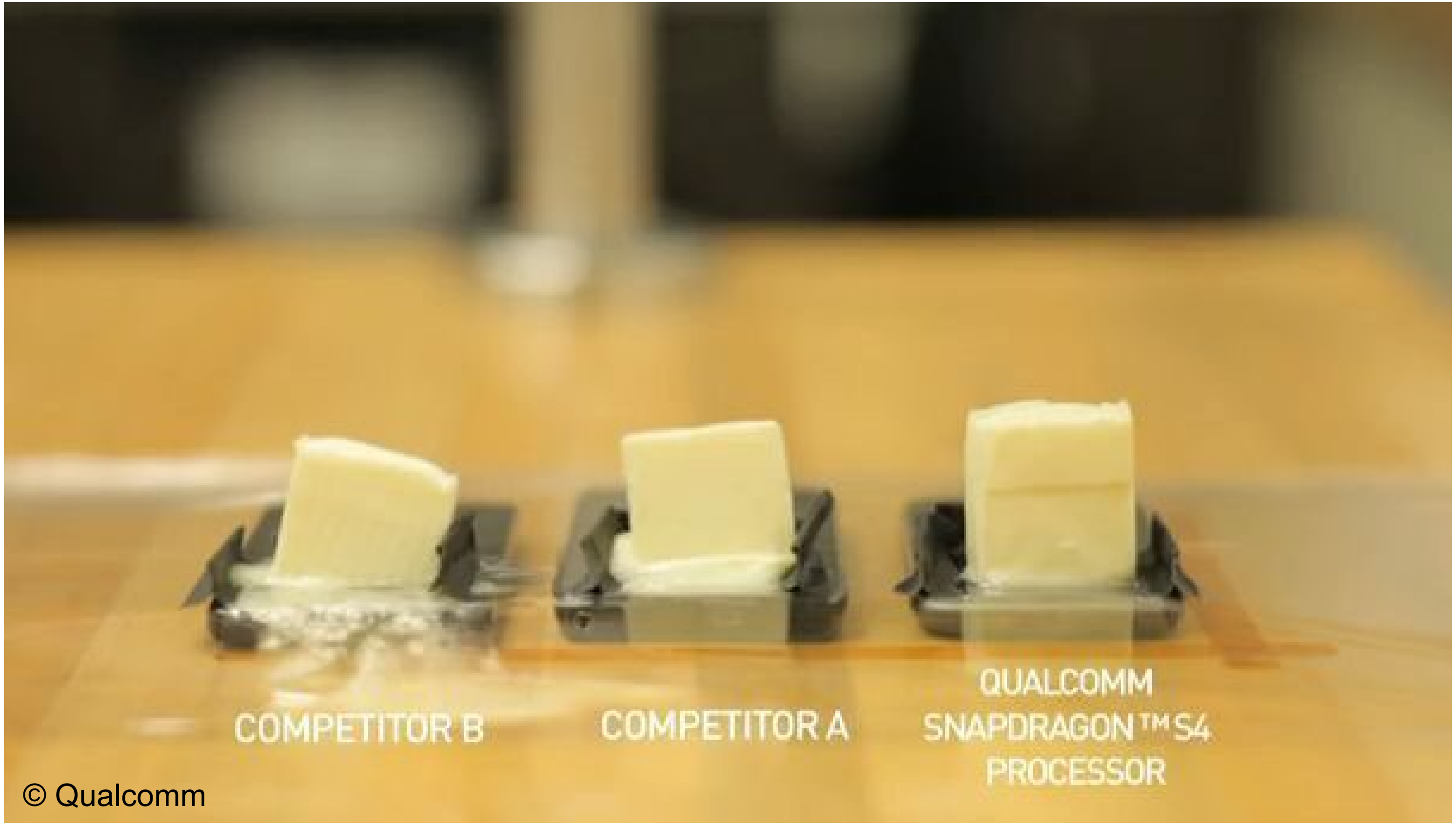
Friedrich-Alexander-Universität
Erlangen-Nürnberg

Power and Energy Consumption of HPC Systems

Georg Hager

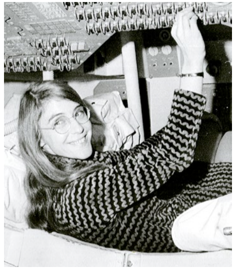
Erlangen National High Performance Computing Center (NHR@FAU)

Points of view: mobile devices



Points of view: computational science

Scientist ("nerd")



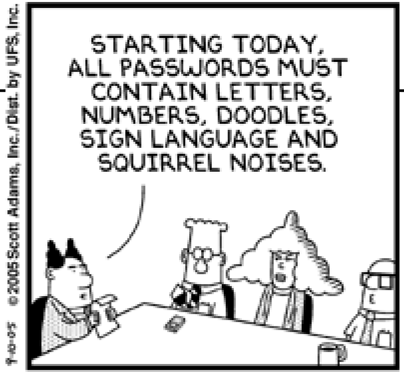
CPU time allocation

Project runtime

Science

Metric: Papers/CPUh

Computing Center ("naggers")



Hardware & maintenance cost

Energy cost

Infrastructure

Machine lifetime

Science

Next machine

Metric: ???

HPC is “hot”

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	1,206.00	1,714.81	22,786
2	**Aurora** - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel DOE/SC/Argonne National Laboratory United States	9,264,128	1,012.00	1,980.01	38,698
3	**Eagle** - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, Microsoft Azure Microsoft Azure United States	2,073,600	561.20	846.84	
4	**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
5	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,752,704	379.70	531.51	7,107

Source: Top500.org, June 2024

Electrical energy at FAU

Origin and consumption of energy at FAU

Approximately 38,000 students and 6,500 members of staff study, research and work in FAU buildings. They require energy, i.e. electricity and heating, in order to carry out their work.

Electricity

Origin

- FAU purchases 100% certified green electricity (renewable)
- A small share of electricity is produced by the university itself from
  - A cogeneration unit
  - Photovoltaic systems

Necessary for

- Lighting
- Ventilation and air conditioning systems
- All devices and appliances (from the office to the laboratory)
- Cooling systems etc.

Consumption

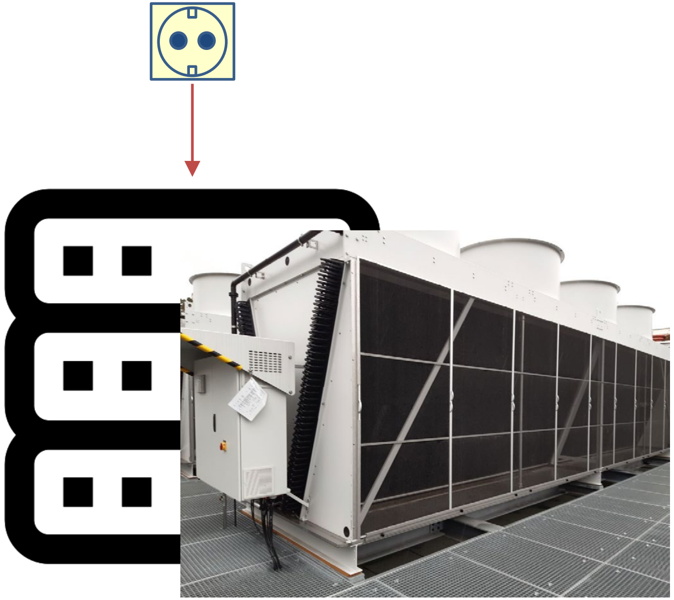
- Approx. 68 GWh per year

https://www.fau.eu/fausavesenergy/infos-zum-energieverbrauch/

Power and energy of a computer system

Energy “consumption” (goes into heat):

$$E = \int_0^T W(t)\,\mathrm{d}t$$



- W: power dissipation [W]
- T: Runtime
- E: Energy [Wh]

≈ 1500

- Example: Fritz cluster at full load 650 kW x 8700 h = 5.7×10^6 kWh p.a.

The price of energy (i.e., why should you care?)

- Energy price ϕ [€/kWh]
- Assuming 0.30 €/kWh, 1 year of Fritz is

$$W \times T \times \phi = 650\ \mathrm{kW} \times 8700\ \mathrm{h} \times 0.30\ €/$$

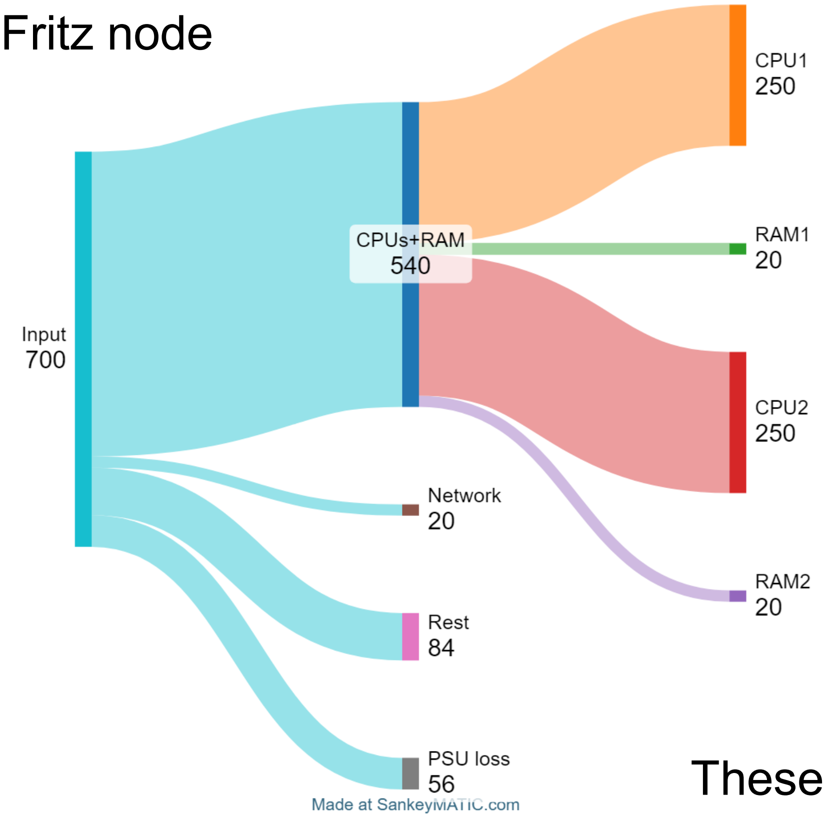
$= 1.7 \times 10^6\ €$ on the energy bill

- How much does my job cost per day?
  - Fritz: 1 node (650 W) for 24 hours $\approx$ 4.70 €
  - Alex: 1 GPU (300 W) + host share (20 W) for 24 hours $\approx$ 2.30 €
  - This does not include the cooling infrastructure
    - + 5-10% for Fritz
    - + 20-30% for Alex

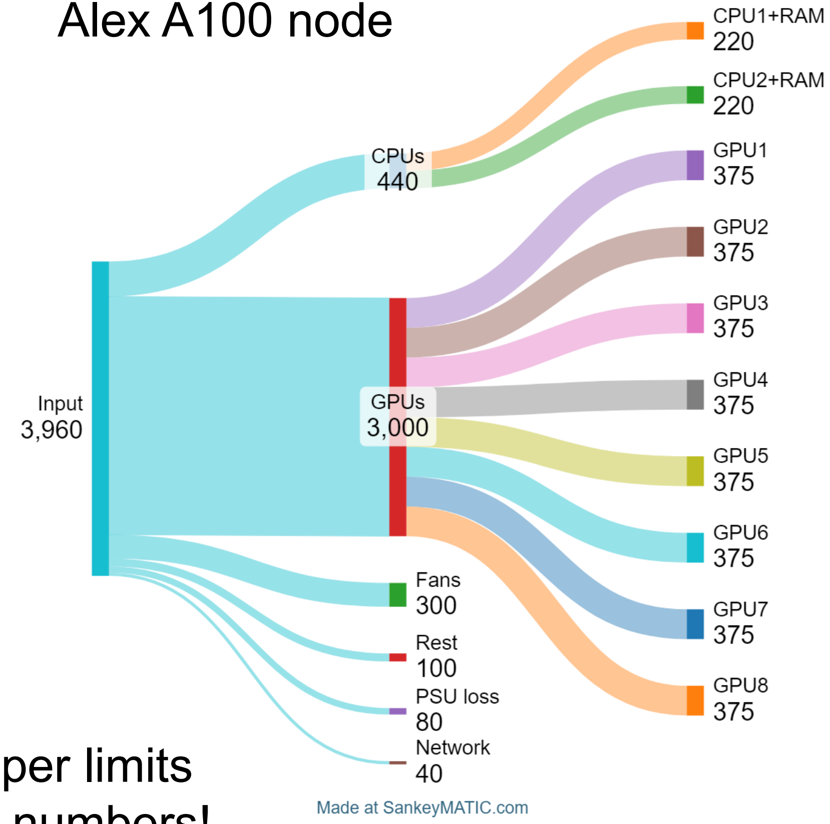
Yes and no…

Where does the power go in a system?

Fritz node



Alex A100 node



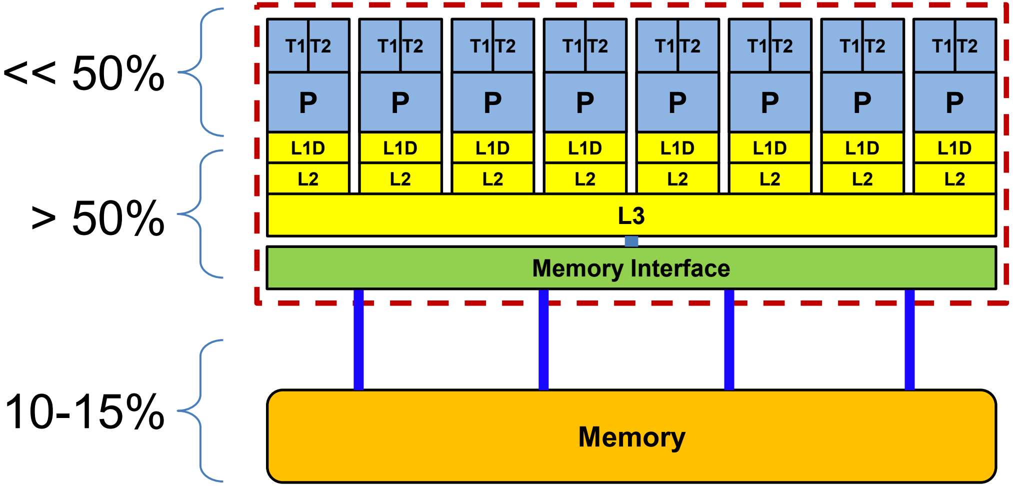
These are upper limits
and eyeballed numbers!

Power dissipation on the chip

- Data transfer is the most energy-intensive thing in a computer system today!
- All else being equal, reducing data transfers reduces the power dissipation
- The further away the data, the more energy it costs

- Good news: Long-distance data lines are also slow and few

- Significant power even for "idle" chip!



How do I know how much power/energy my job uses?

- Tools (CPU)
  - LIKWID tools (based on RAPL)
    - Available as module on NHR@FAU clusters
    - https://github.com/RRZE-HPC/likwid
    - `likwid-perfctr -g ENERGY [-m] -c N:0-71 ./a.out`
    - `likwid-mpirun -g ENERGY [-m] -np 360 ./a.out`
  - ClusterCockpit (based on LIKWID)
    - https://github.com/ClusterCockpit

- Tools (GPU)
  - Nvidia + AMD tools `nvidia-smi` / `rocm-smi`
  - ClusterCockpit (based on `nvidia-smi`)

ClusterCockpit

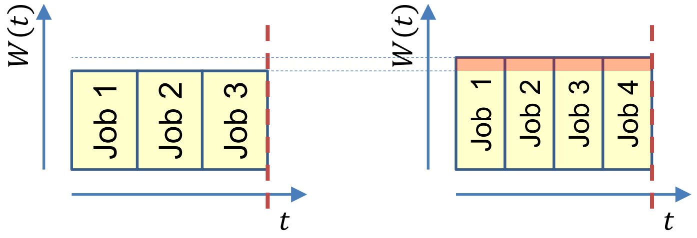
Job-specific monitoring accessible from HPC portal

Job energy consumption will be available soon™

Reducing the energy per job

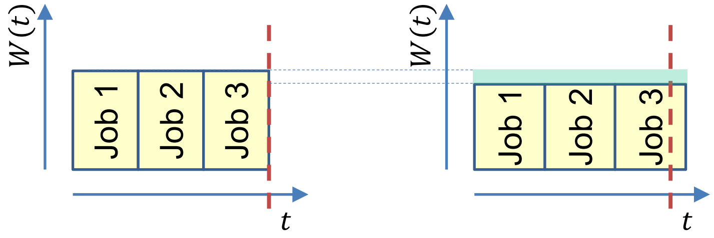
Reduce the job's runtime T

- Better overall use of resources
- Better use of your resource allotment
- More "science per CPUh"
- Probably higher or lower power dissipation of the system

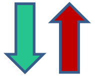


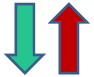
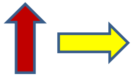
Reduce the job's power $W(t)$

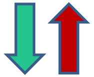
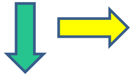
- Lower power dissipation of the system
- Probably some performance loss → probably less "science per CPUh"



Tuning knobs for job energy reduction (user view)

- Code performance (“optimize code”)
  - Power  Energy  CPU/GPU-h 

- Clock speed reduction (“underclocking”)
  - Power  Energy  CPU/GPU-h 

- Concurrency throttling (“use less hardware”)
  - Power  Energy  CPU/GPU-h 

Summary: It’s complicated.

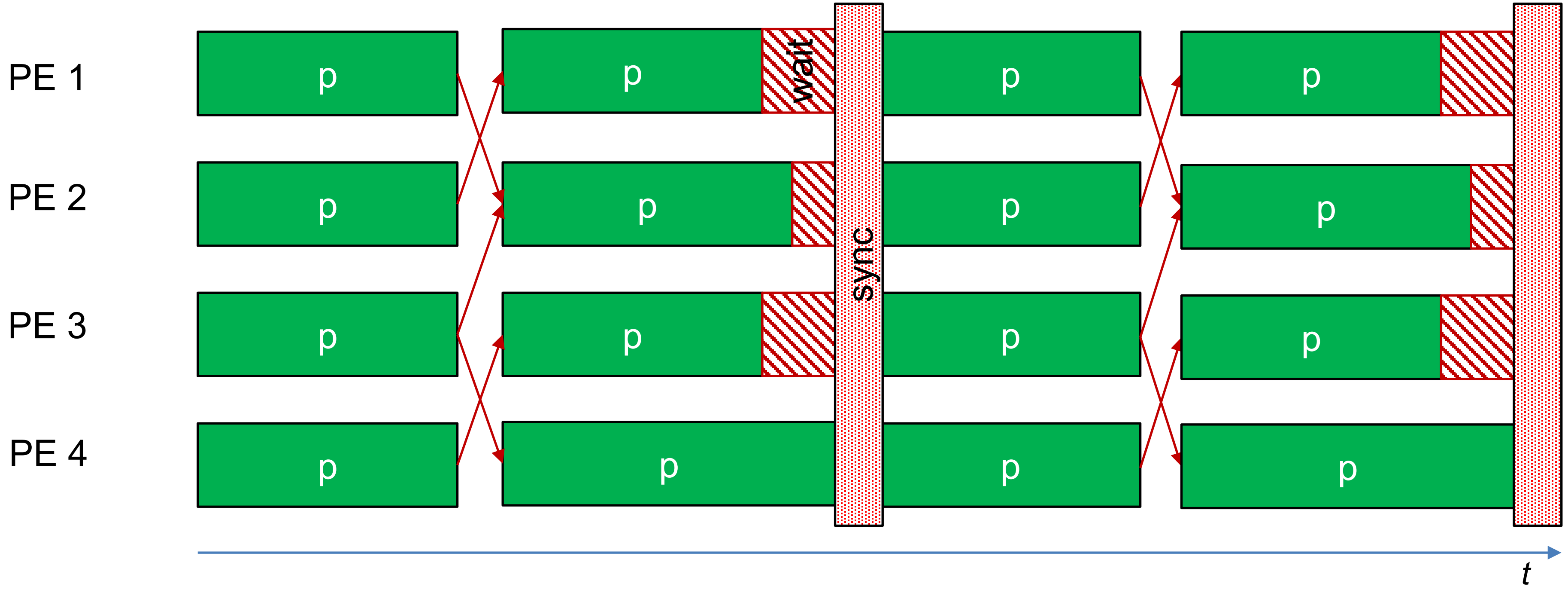
Code optimization for runtime and energy

- Use the best algorithm
  - E.g., $N^2 \rightarrow N \log N$
- Use optimized libraries
  - E.g., OpenBLAS → MKL
- Use aggressive compiler optimization
  - E.g., `-O1` → `-Ofast -xHost`
- Do less work
  - E.g., use sparse matrices instead of dense

- Balance the workload
  - All “devices” finish at the same time
- Transfer less data from far away
  - Cache blocking / register reuse
  - Avoid network communication
- Hide communication overhead
  - Asynchronous/bidirectional network communication
  - Asynchronous GPU data transfers

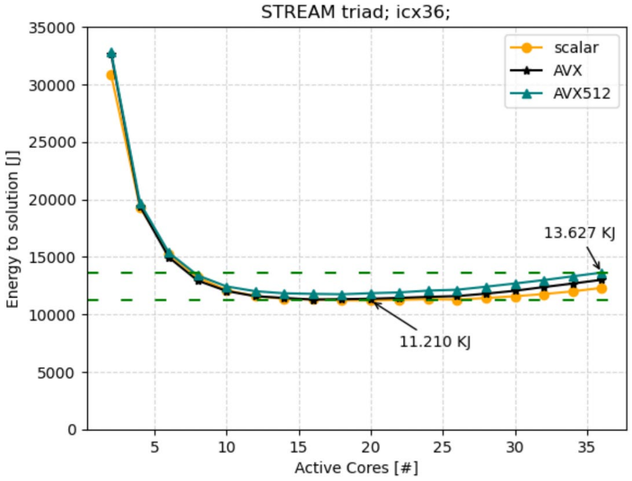
Concurrency and resource efficiency

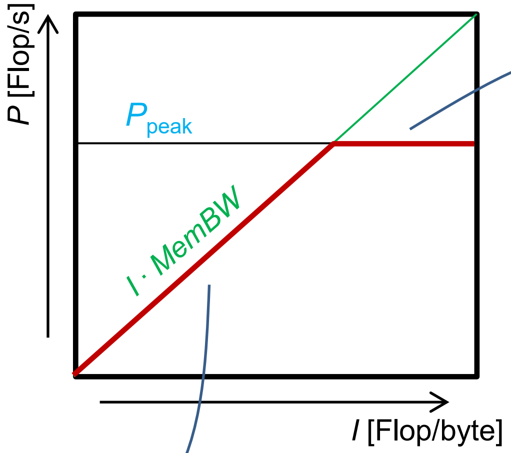
- In general, less parallelism is better for resource efficiency
- Parallelism incurs overhead, which is waste



… unless we have relevant shared resources

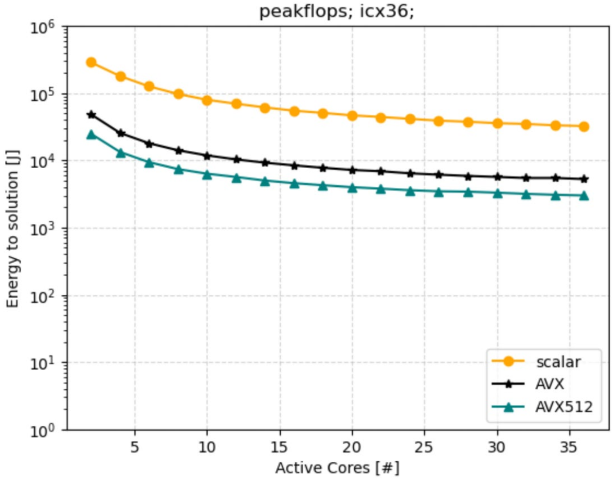
Roofline point of view on a CPU socket level





memory bound
(“saturating”)

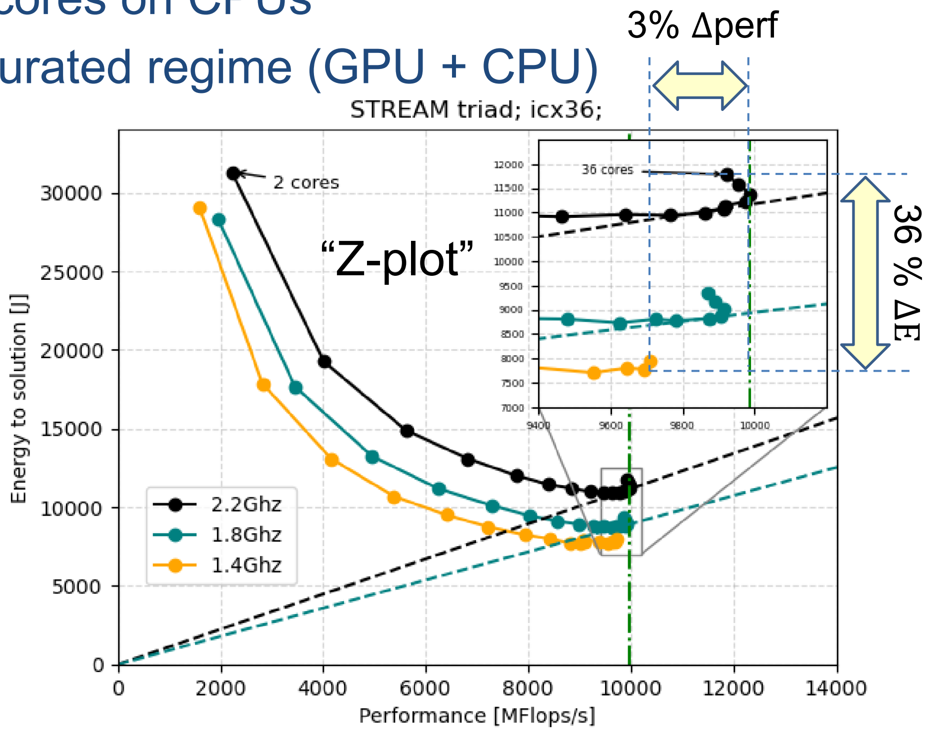
core bound
(“scalable”)



Data: Master thesis Stefan de Souza

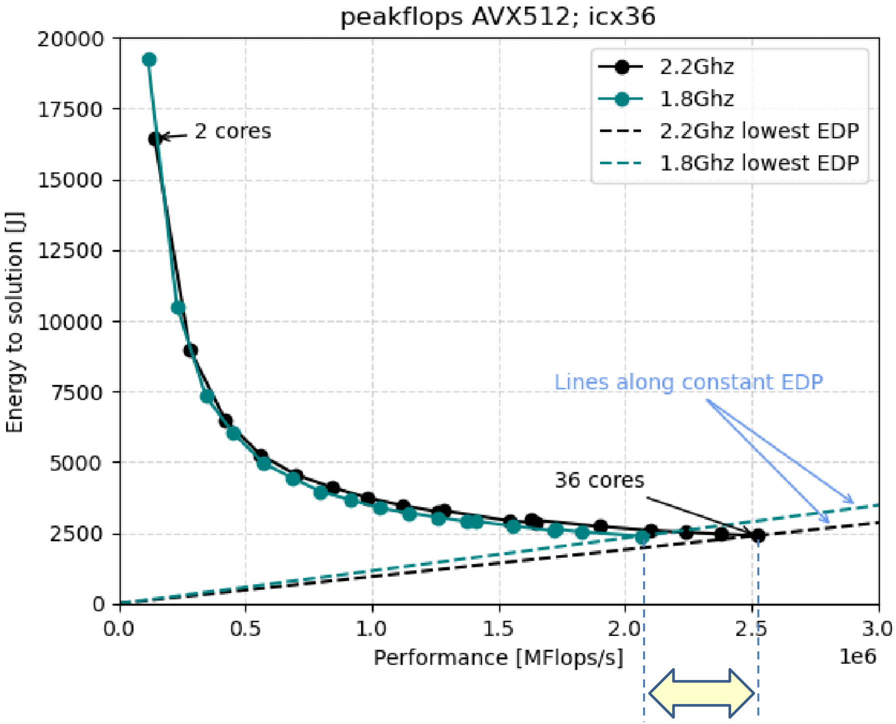
Memory-bound code

- Shows "performance saturation" vs. # of cores on CPUs
- Weak sensitivity to clock frequency in saturated regime (GPU + CPU)
  - Significant energy saving potential
- Optimal "operating point" is crucial
  - # of cores at saturation point → min E
  - Clock speed
- Setting the concurrency
  - `OMP_NUM_THREADS`
  - `-nperdomain`
  - … whatever, but use pinning!
- Setting the clock speed (CPUs)
  - `likwid-setFrequencies -f <GHz>`
  - `srun --cpu-freq=<kHzmin>-<kHzmax>:performance ./a.out`



Core-bound ("scalable") code

- No on-chip scaling bottleneck
- Performance scales with # cores
- The more cores, the lower the energy to solution
- Performance sensitive to clock speed
  - Ideally, proportional
- Power is sensitive to clock speed
  - $W \propto f^{\alpha}, \alpha \in [1, \ldots, 3]$
  - There is an optimal frequency for minimal energy to solution
  - … at the price of significant performance loss



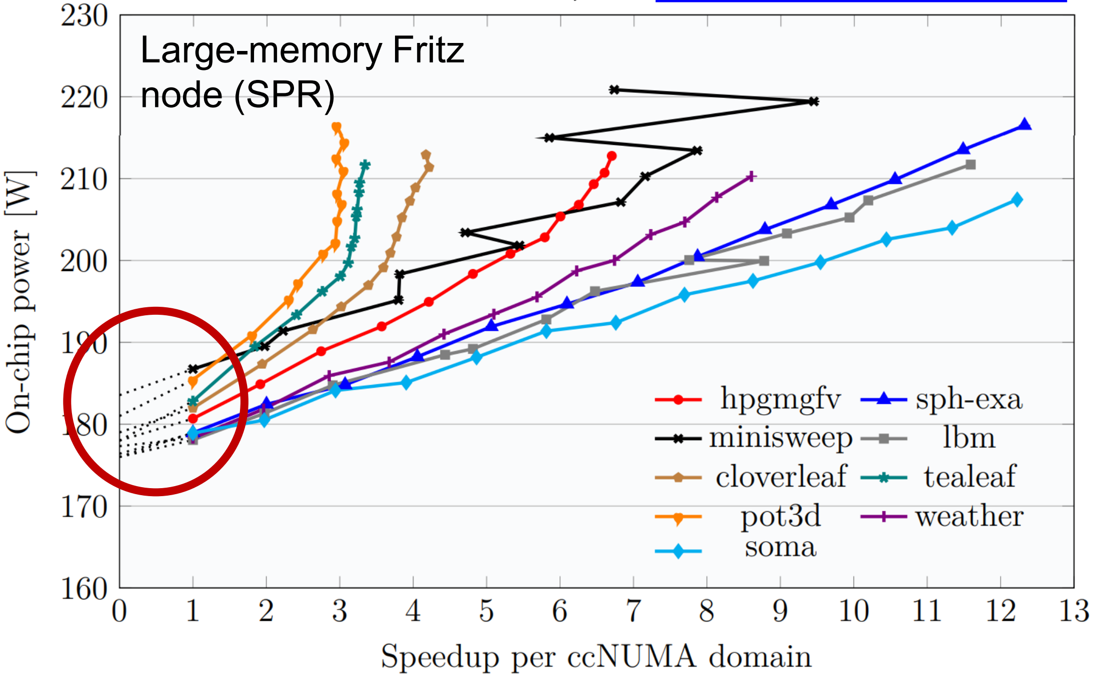
10% Δperf
@ zero ΔE

Chip baseline power today

- Much of the CPU/GPU chip power today is “baseline power”
- Extrapolation to “zero concurrency”

Afzal et al., DOI [10.1145/3624062.3624197](https://doi.org/10.1145/3624062.3624197)

- Fritz:
  - ICL socket $W_0 \approx 100$ W (TDP = 250W)
  - SPR socket $W_0 \approx 180$ W (TDP = 350W)
- Sandy Bridge (2012): 20% baseline power



What can the computing center do?

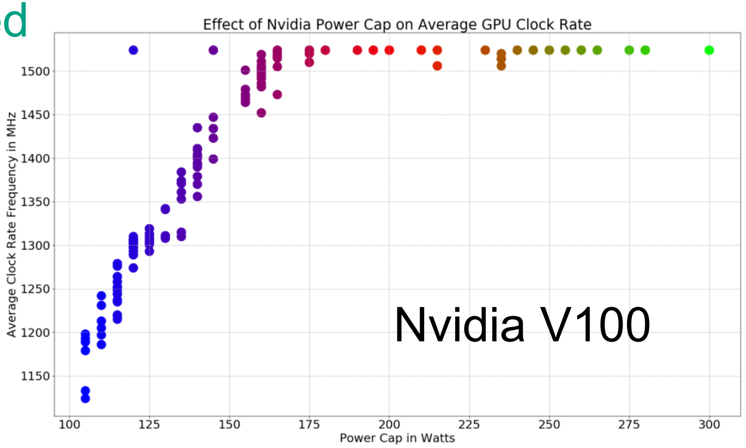
Disable Turbo Mode, fixed frequency

- Reliable performance behavior
- Damped but still significant variations in power
- Pessimistic power power limit

- Infrastructure must provide power headroom

Device power capping

- Reliable upper power limit
- Power variations turn into performance variations
- Infrastructure can be tightly power limited



Patki et al., DOI: [10.1109/WORKS49585.2019.00009](https://doi.org/10.1109/WORKS49585.2019.00009)

So… what should you do?

0th order advice:

- Make your code run faster
  - Latest code versions
  - Best/latest compilers
  - Best libraries
  - Code optimization

- Avoid waste
  - Reasonable concurrency
  - Controlled overhead
  - “If it finishes in tolerable time, don’t scale further”

For memory-bound code:

- Significant energy savings via clock speed reduction
  - Run benchmarks
  - Talk to your friendly computing center

- Consider applying for a KONWIHR project
  - www.konwihr.de

NHR FAU

Friedrich-Alexander-Universität Erlangen-Nürnberg

Thank you.