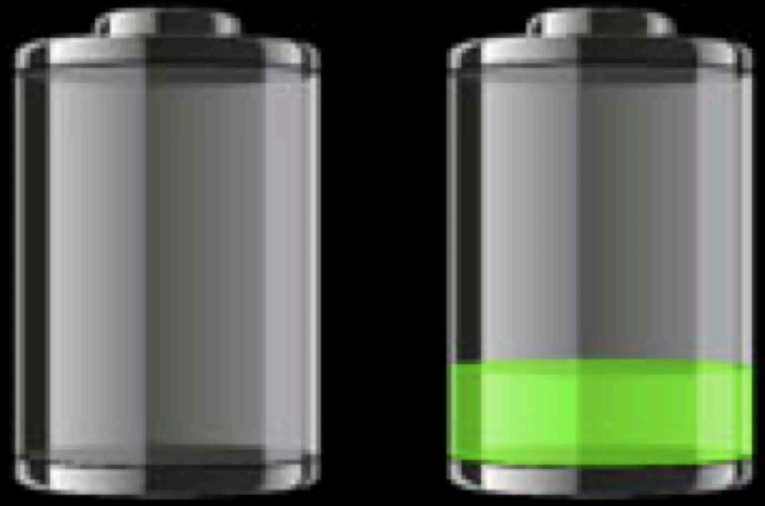
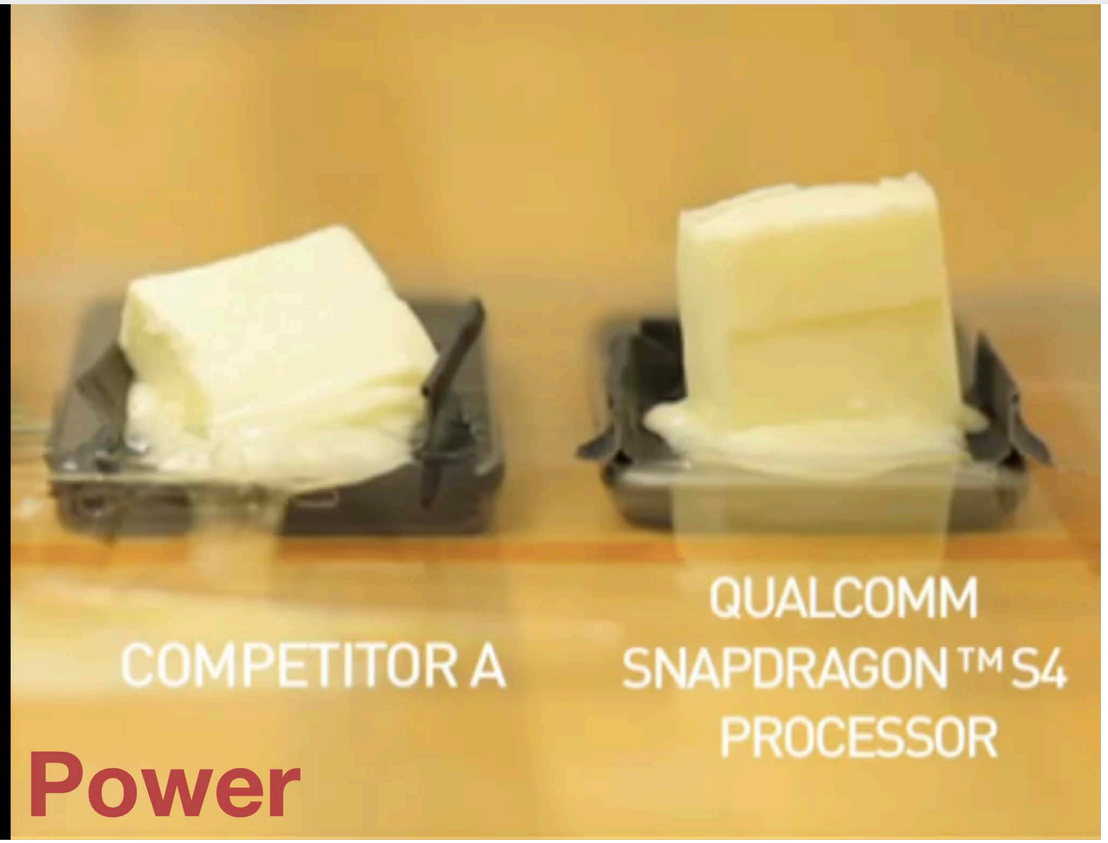
Rich Vuduc, Georgia Tech

What is an energy-efficient algorithm?





Energy **Power**

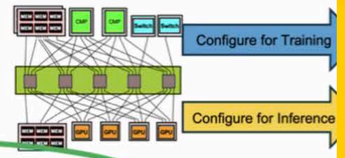
Rich Vuduc (... | Zoom Facilita... | Presentations | Meeting Room | Jerry Bernholc

Rich Vuduc (Georgia... | Zoom Facilitator 1 | Presentations | Jerry Bernholc

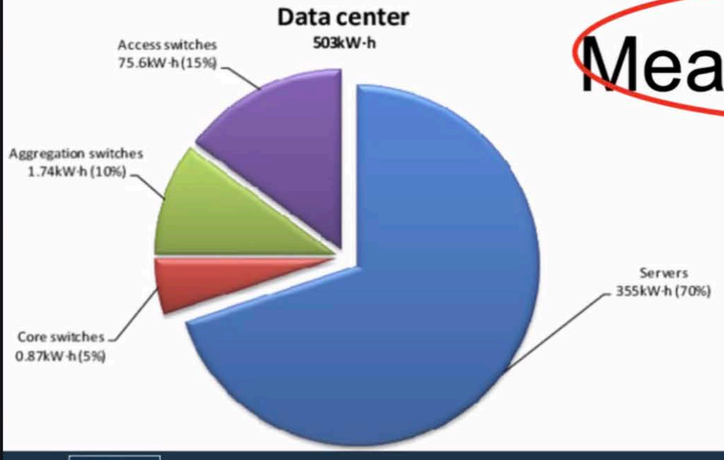
Anatomy of a "Value" Metric

Increase performance with Disaggregation
And Bandwidth Steering!

Deliver bandwidth and resources to where it is needed
By taking it from where it isn't



$$\frac{\text{Performance}}{\text{Measured Watt}}$$



30% of datacenter power goes to network

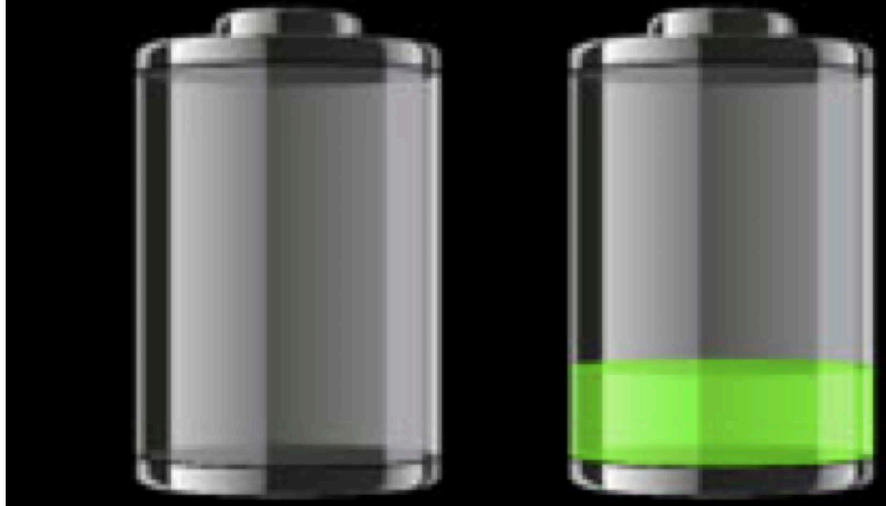
So max savings by creating a perfectly efficient (0 pJ/bit) optical interconnect is ONLY 30%!

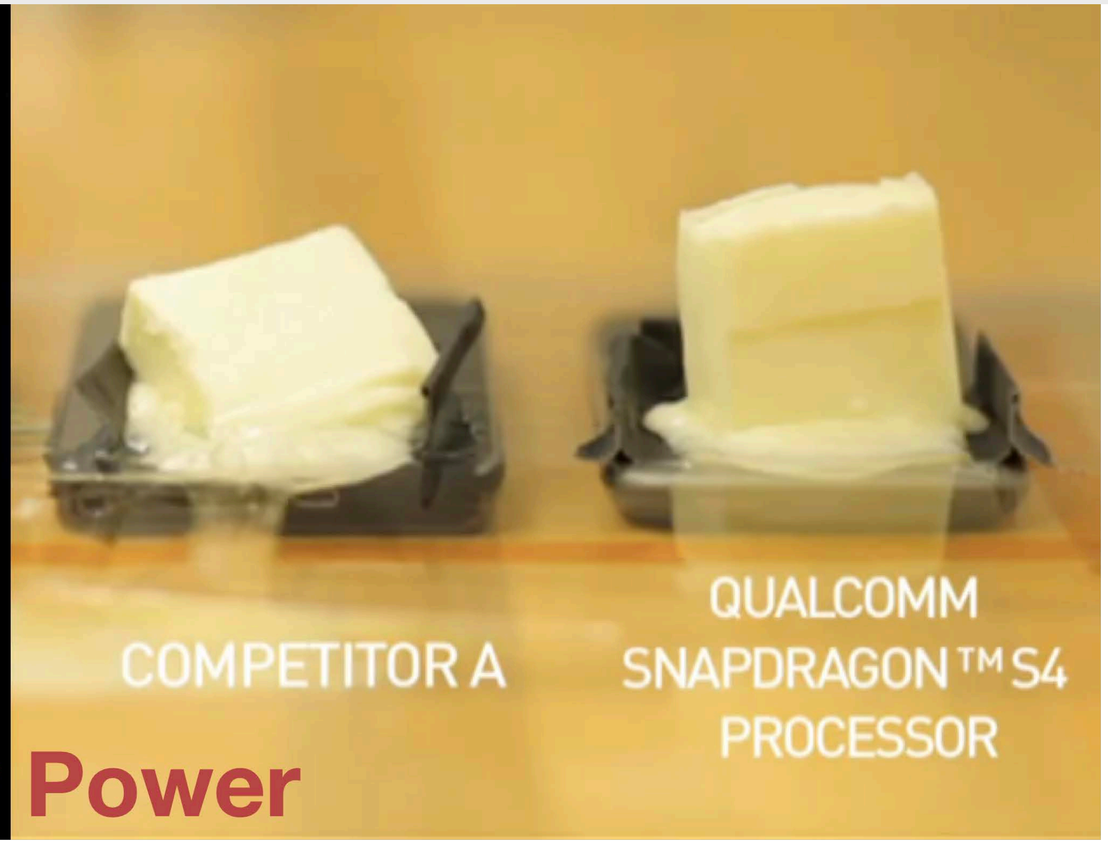
BERKELEY LAB

Exploit the unique capabilities to improve both numerator and denominator!

Rich Vuduc, Georgia Tech

Can you beat the “standard” model?





Energy **Power**

What is the “standard model?”

What can it tell me about building energy-efficient systems?

Recall: Sequential complexity analysis

$$\mathcal{O}(N^2) \longrightarrow \boxed{\mathcal{O}(N)}$$

Count asymptotic operations & **reduce them**.

Recall: Sequential complexity analysis

$$\mathcal{O}(N^2) \longrightarrow \boxed{\mathcal{O}(N)}$$

What about energy?

Recall: Sequential complexity analysis

$$\mathcal{O}(N^2) \longrightarrow \boxed{\mathcal{O}(N)}$$

Reduces **energy**, e.g., fewer ops, less storage

The “classical process” ***is*** energy reduction!

Recall: Sequential complexity analysis

$$\mathcal{O}(N^2) \longrightarrow \boxed{\mathcal{O}(N)}$$

But if that's so, what does **explicit reasoning** about energy accomplish?

Recall: Sequential complexity analysis

$$\mathcal{O}(N^2) \longrightarrow \boxed{\mathcal{O}(N)}$$

Answer: It **facilitates energy-centric codesign** and will **make or break** your new platform.

$$\mathcal{O}(N^2) \longrightarrow \mathcal{O}(N)$$

What is an “**operation**?”

$$\mathcal{O}(N^2) \longrightarrow \mathcal{O}(N)$$

It is an “**interface**” that enabled >70 years of **productivity** in algorithms (and software).

$$\mathcal{O}(N^2) \longrightarrow \mathcal{O}(N)$$

An implication? Your new computing substrate abandons this interface at its **peril**.

$$\mathcal{O}(N^2) \longrightarrow \mathcal{O}(N)$$

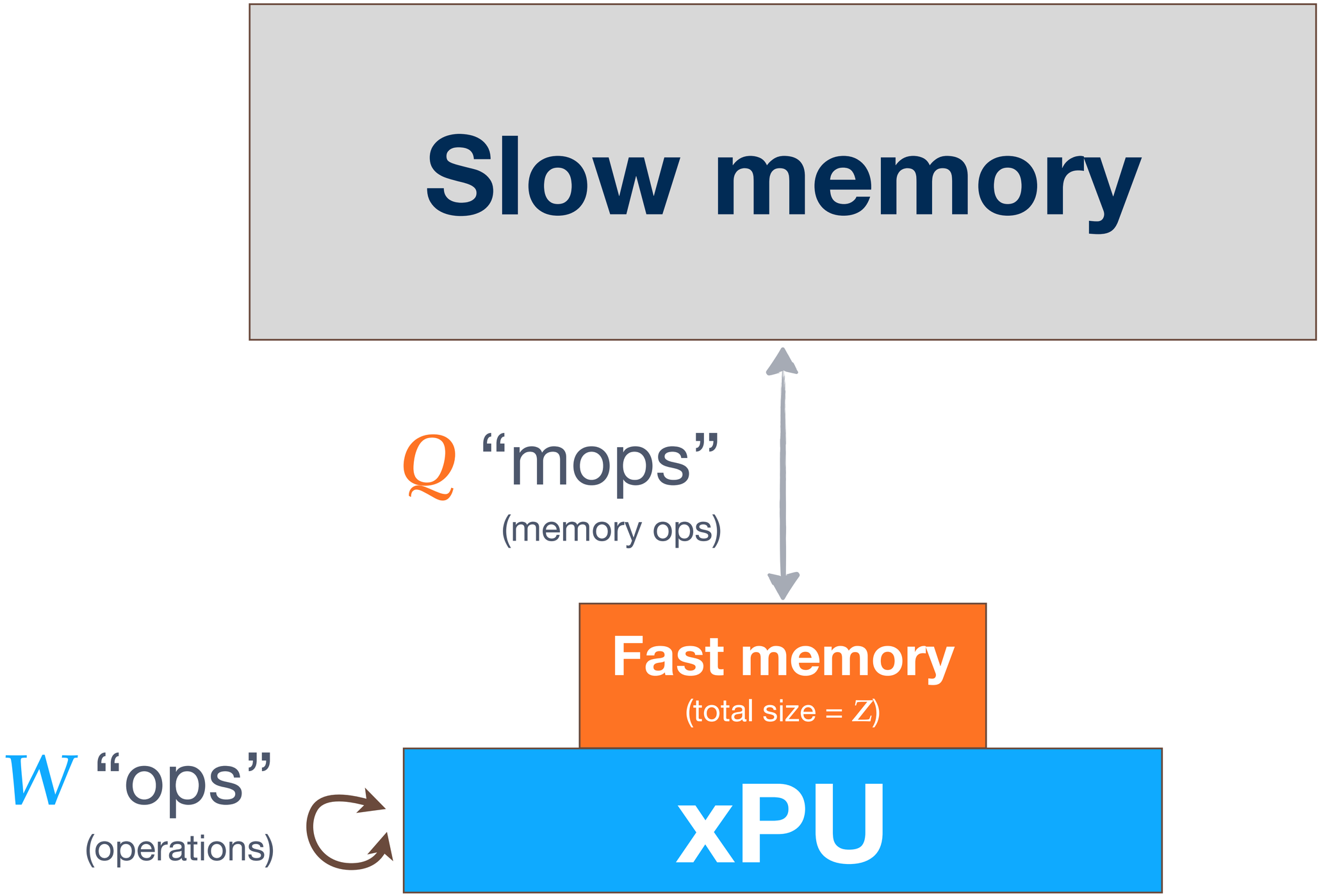
What else does an “**operation**” do, in its role as an interface?

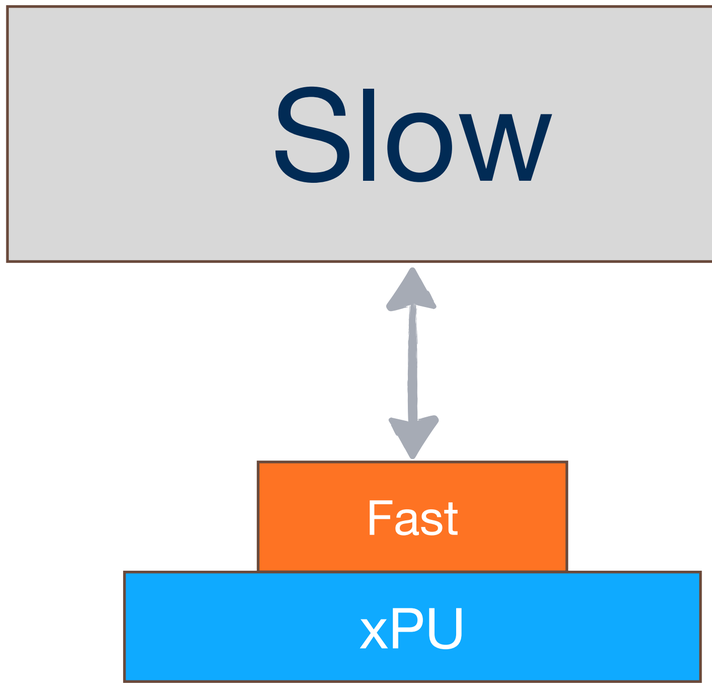
$$\mathcal{O}(N^2) \longrightarrow \mathcal{O}(N)$$

It defines a **unit of cost**.

$$\mathcal{O}(N^2) \longrightarrow \boxed{\mathcal{O}(N)}$$

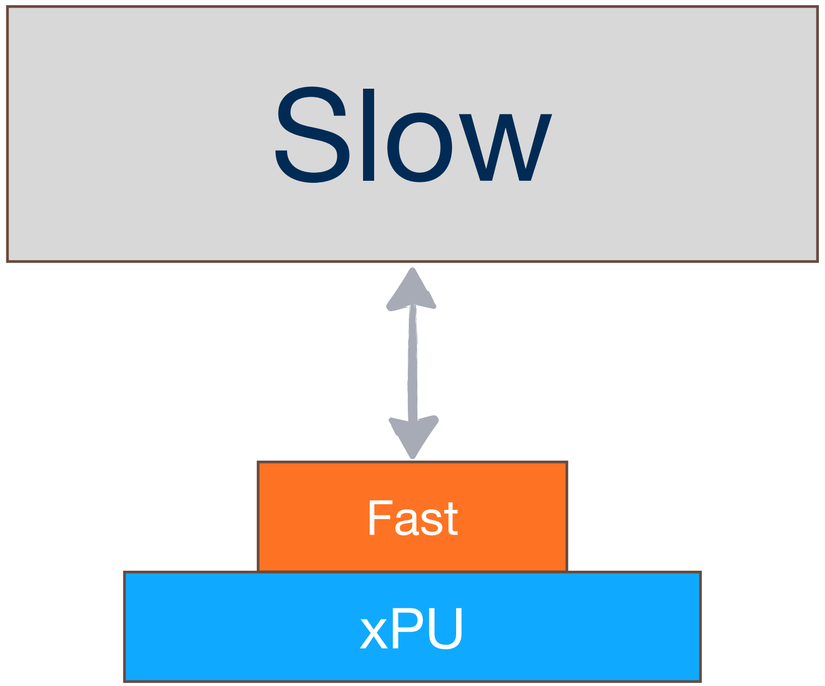
It defines a **unit of cost**. “Exciting” things happen when costs are **nonuniform**.


Slow memory
Q “mops”
(memory ops)
Fast memory
(total size = Z)
W “ops”
(operations)
xPU



W **ops**

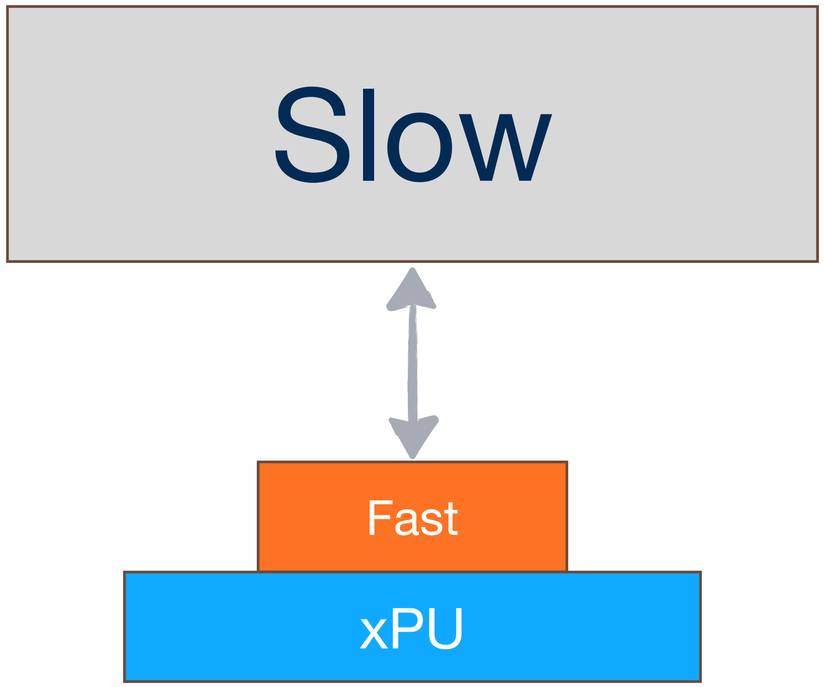
Q **mops**



$$\frac{W}{Q}$$

op : mop

(e.g., “flop:Byte”)



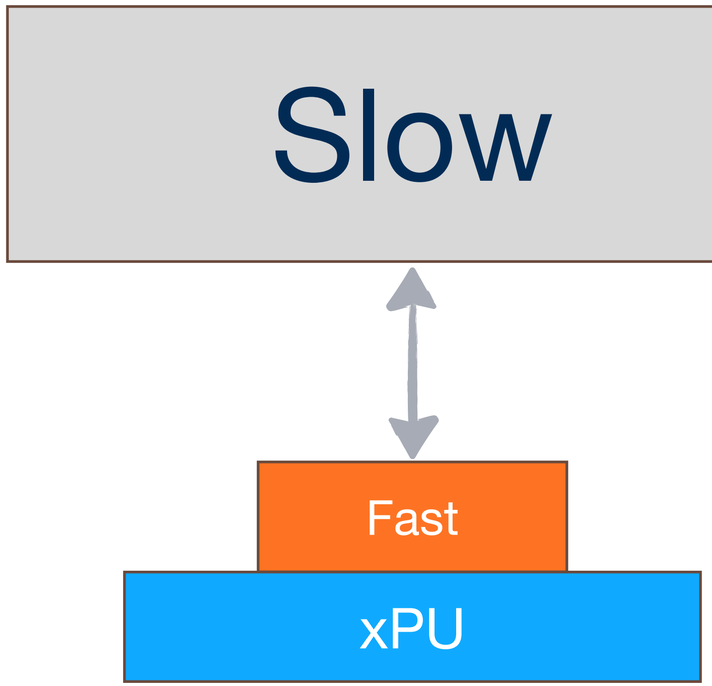
Time per op

$$W \tau_{\text{op}}$$

(In truth, inverse throughput)

$$Q \tau_{\text{mop}}$$

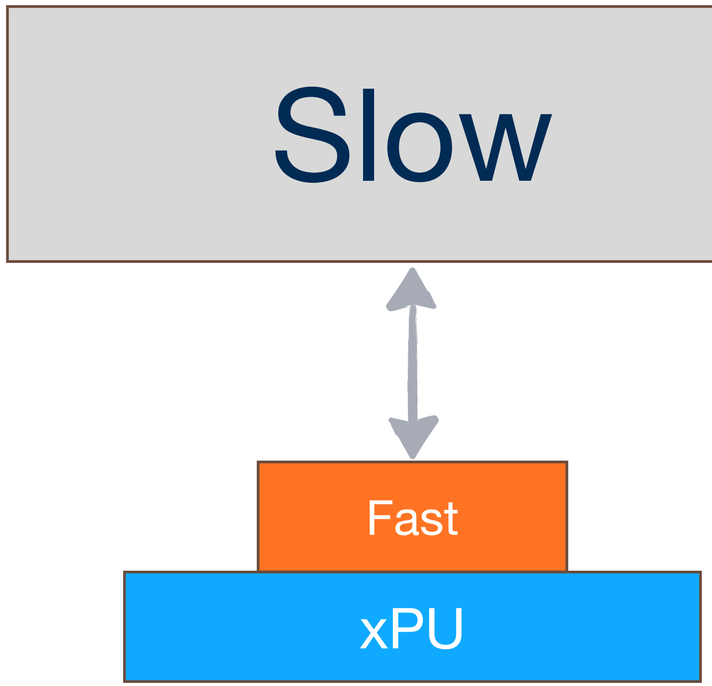
Time per mop



Machine balance
(inverse)

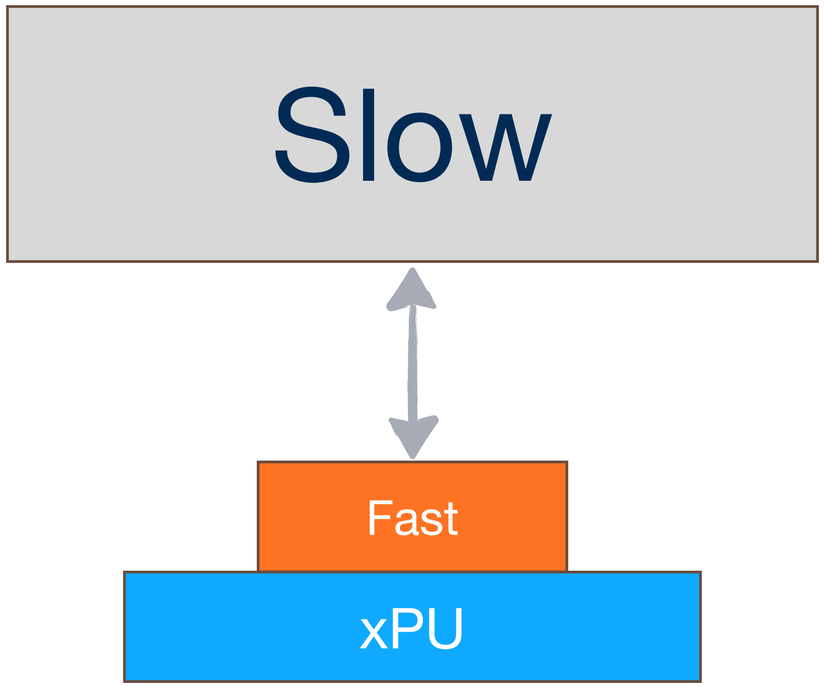
$$\frac{W}{Q} \frac{\tau_{\text{op}}}{\tau_{\text{mop}}}$$

Balance & intensity have a long history: **Hockney (1989)**; Kung (1986); Hillis (1987); Callahan & Kennedy (1988); Blelloch, Maggs, Miller (1994); McCalpin (1995); Williams et al. (2009); …



$$\frac{W\ ??}{Q\ ??}$$

What about **energy?**



Energy per op

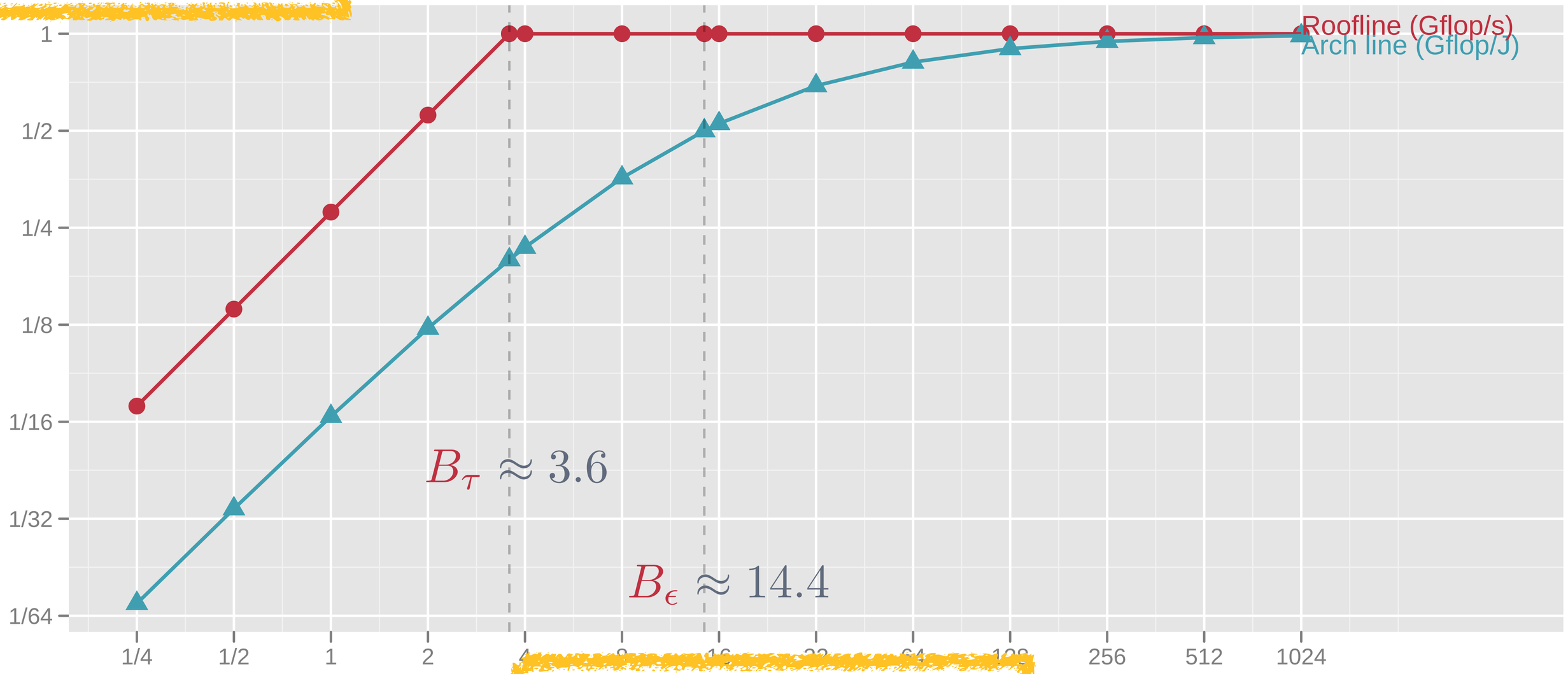
$$\frac{W}{Q}\ \frac{\varepsilon_{\text{op}}}{\varepsilon_{\text{mop}}}$$

Energy per mop

Rooflines in **time** & (igloos in?) **energy**

S. Williams, A. Waterman, D. Patterson. *Roofline: An insightful visual performance model for multicore architectures. CACM*, v52(4), 2009. doi:10.1145/1498765.1498785
J. Choi, D. Bedard, R. Fowler, R. Vuduc. *A roofline model of energy.* In *IPDPS'13*. doi:10.1109/IPDPS.2013.77
J. Choi, M. Dukhan, X. Liu, R. Vuduc. *Algorithmic time, energy, and power on candidate HPC building blocks.* In *IPDPS'14*. doi:10.1109/IPDPS.2014.54

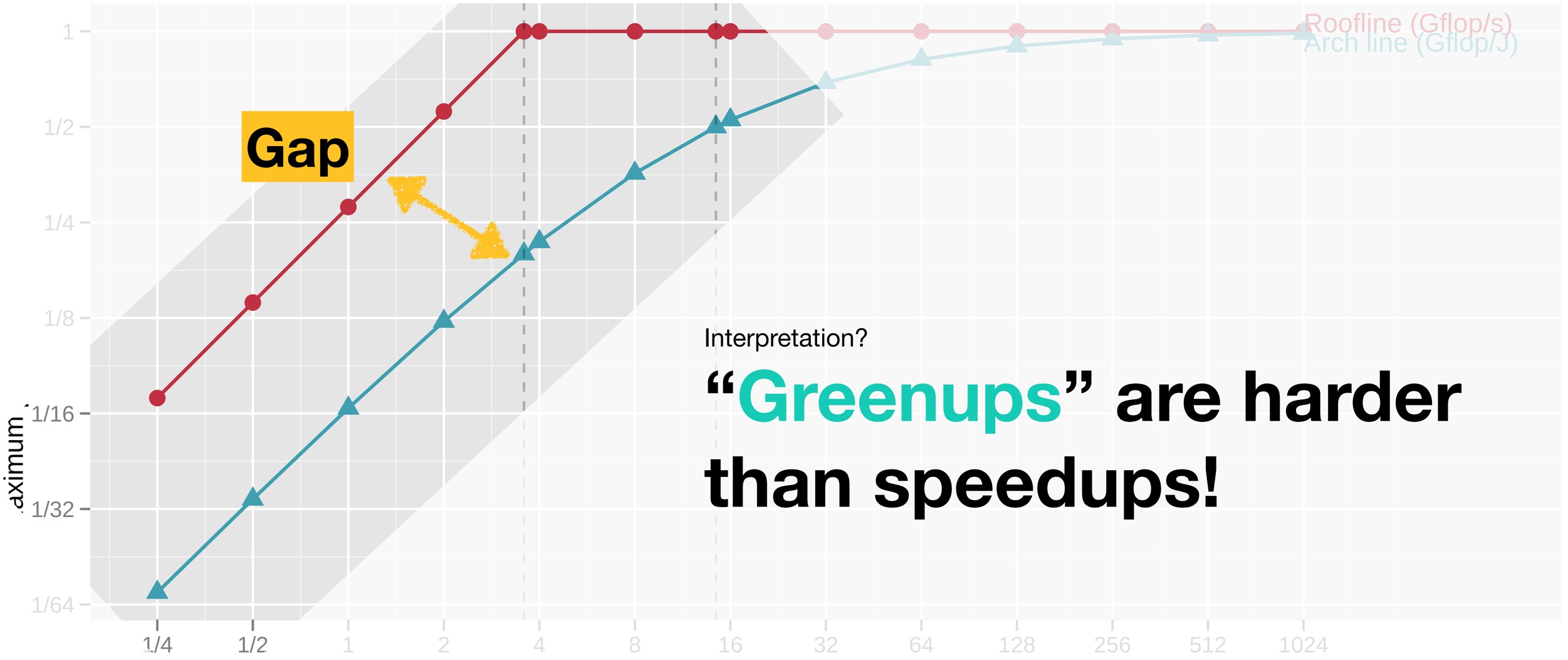
Fraction of peak



Algorithmic (fl)op:byte

Balance estimates for a high-end NVIDIA Fermi, according to Keckler et al. *IEEE Micro* (2011)

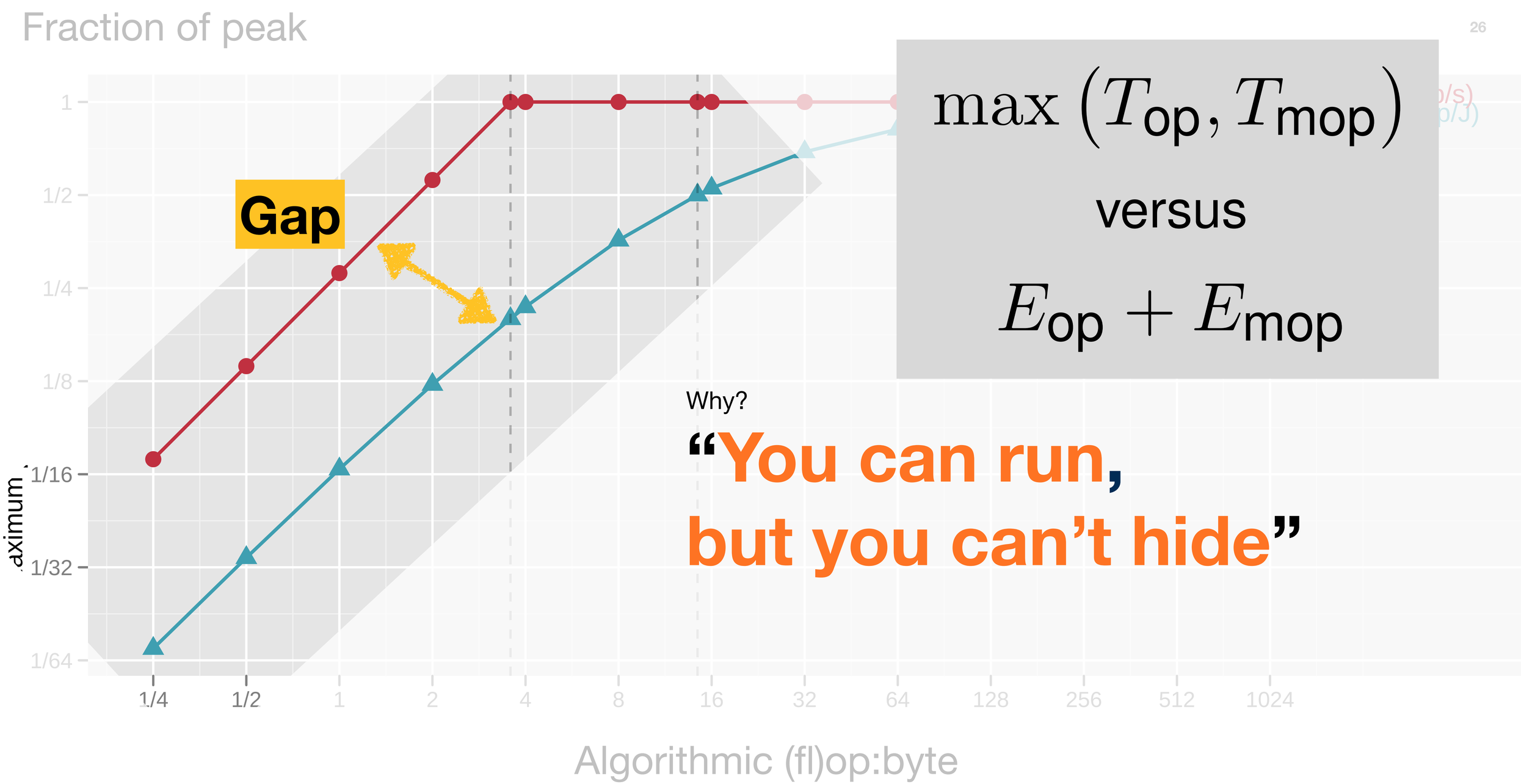
Fraction of peak



Interpretation?

“Greenups” are harder than speedups!

Algorithmic (fl)op:byte



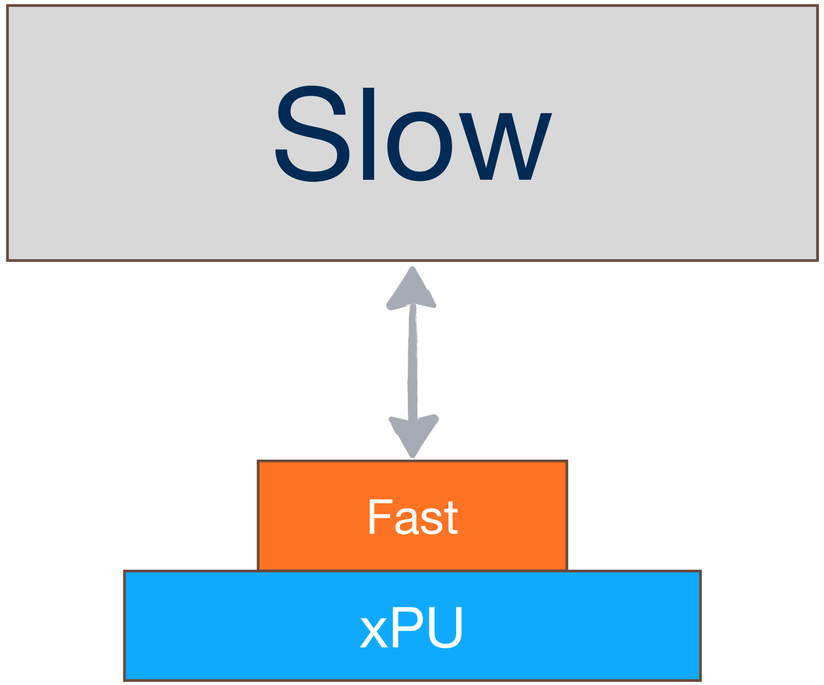
$$\max\left(T_{\text{op}}, T_{\text{mop}}\right)$$

versus

$$E_{\text{op}} + E_{\text{mop}}$$

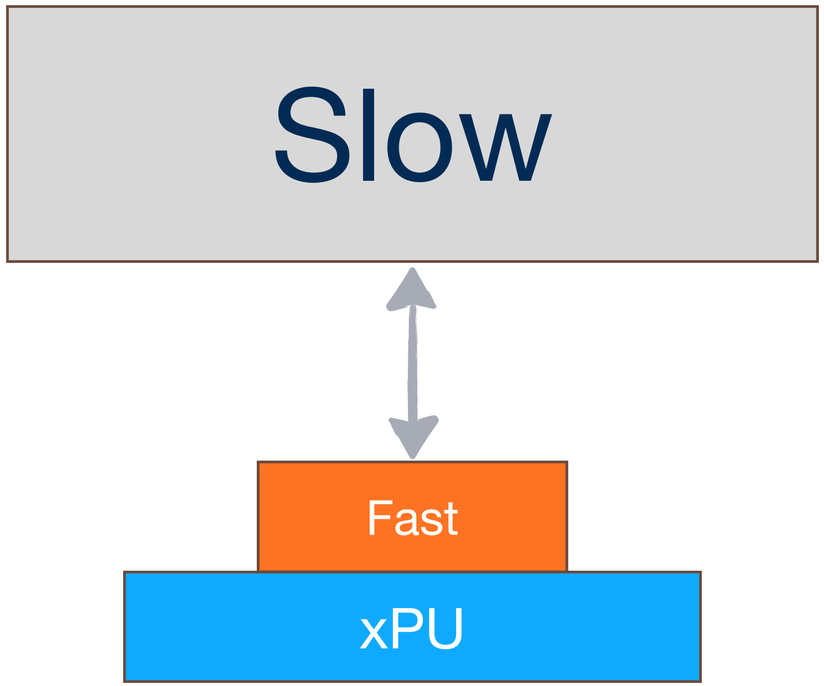
Why?

“You can run, but you can’t hide”



$$\frac{W\ ??}{Q\ ??}$$

What about **power?**

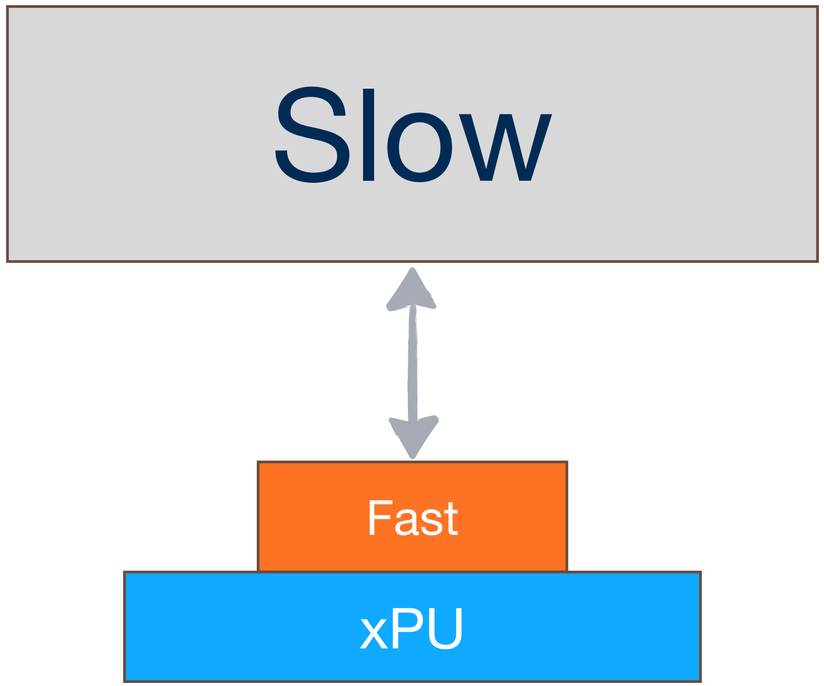


Power per op

$$\frac{W}{Q}\,\frac{\varepsilon_{op}/\tau_{op}}{\varepsilon_{mop}/\tau_{mop}}$$

Power per mop

What about **power?**



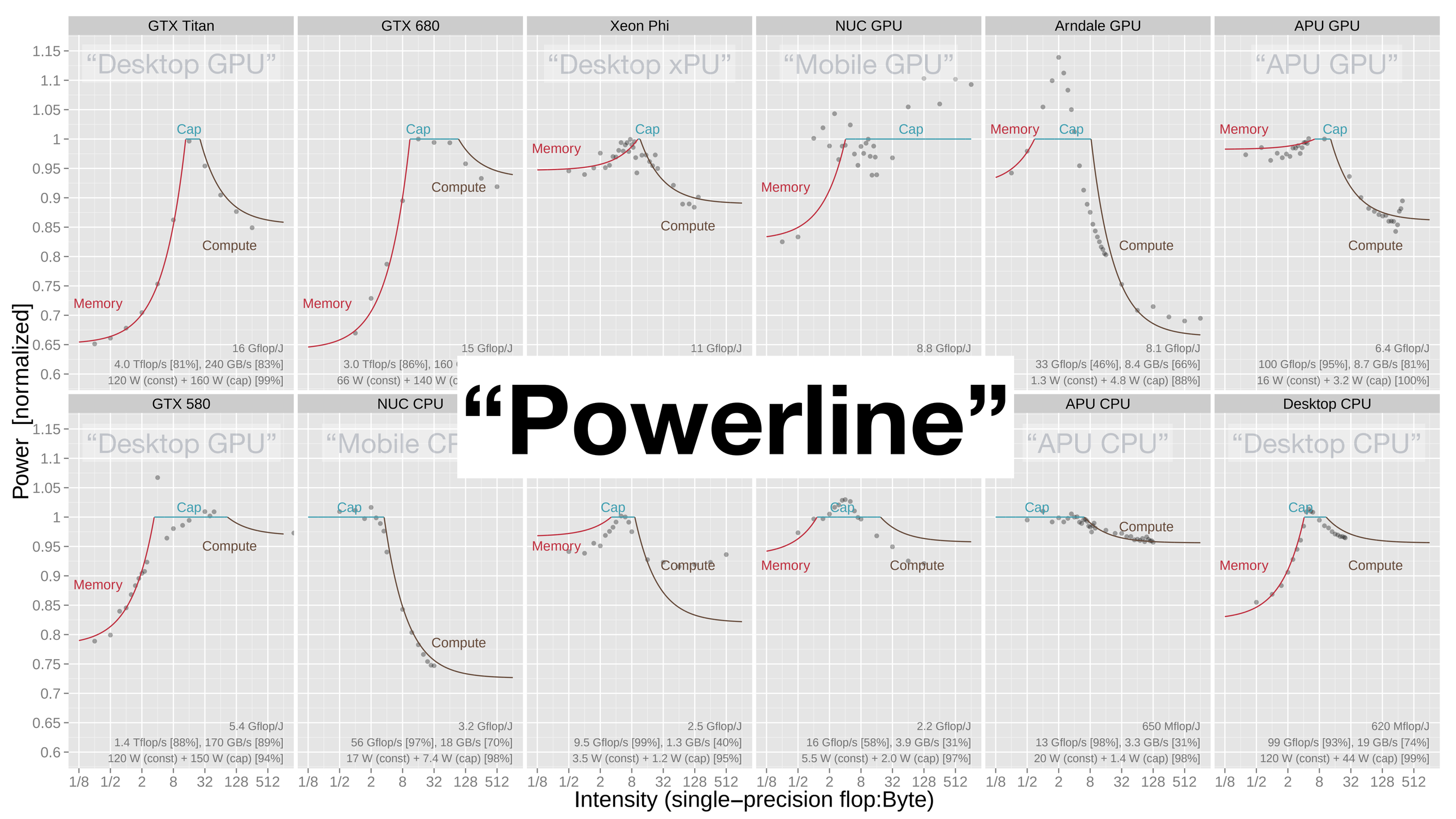
Power per op

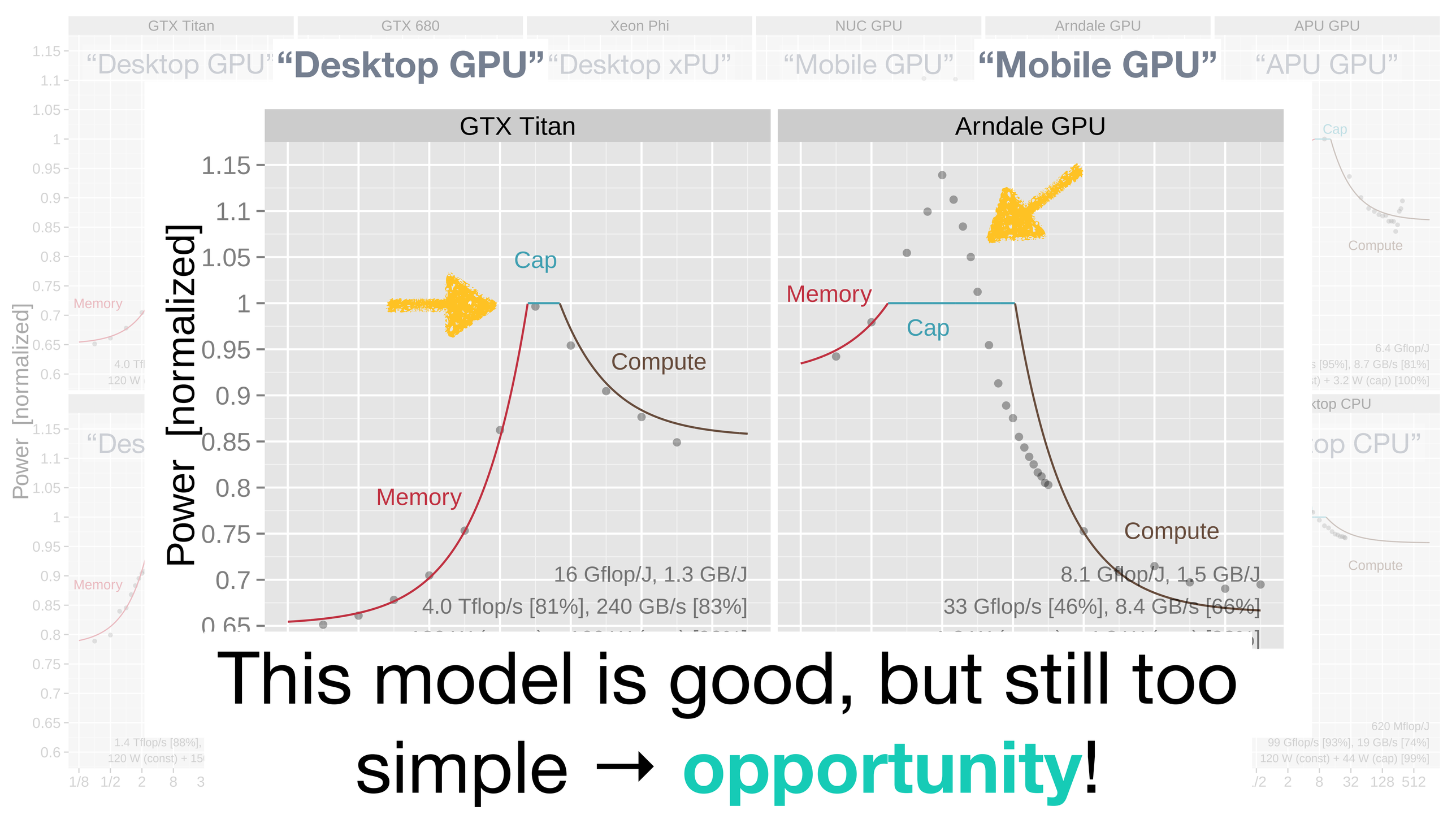
$$\frac{W \quad \varepsilon_{op}/\tau_{op}}{Q \quad \varepsilon_{mop}/\tau_{mop}}$$

Power per mop

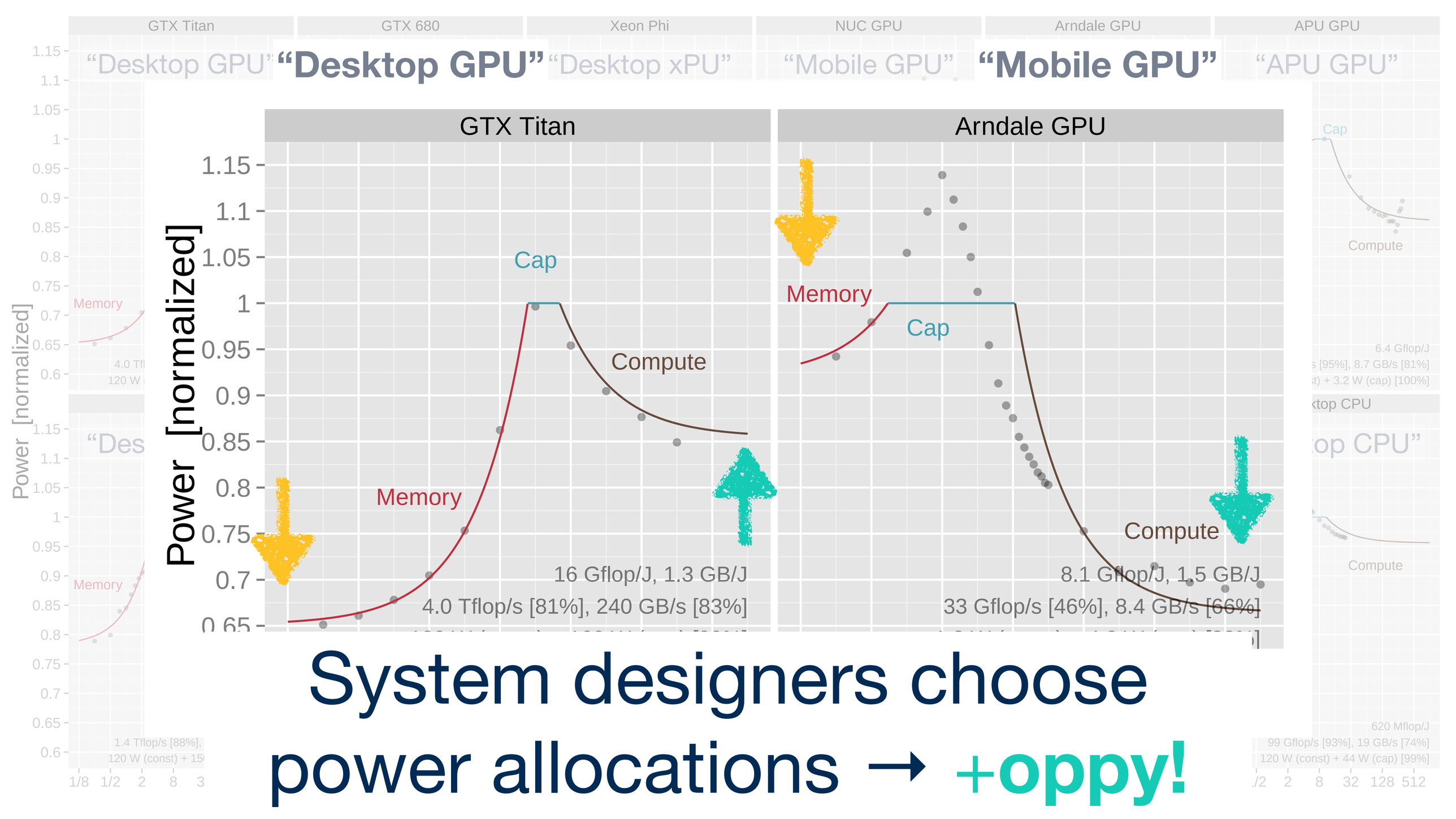
Real-life complications:

Baseline power & power caps


GTX Titan
"Desktop GPU"
Cap
Compute
Memory
16 Gflop/J
4.0 Tflop/s [81%], 240 GB/s [83%]
120 W (const) + 160 W (cap) [99%]
GTX 680
"Desktop GPU"
Cap
Compute
Memory
15 Gflop/J
3.0 Tflop/s [86%], 160
66 W (const) + 140 W (c
Xeon Phi
"Desktop xPU"
Cap
Memory
Compute
11 Gflop/J
NUC GPU
"Mobile GPU"
Cap
Memory
8.8 Gflop/J
Arndale GPU
Memory
Cap
Compute
8.1 Gflop/J
33 Gflop/s [46%], 8.4 GB/s [66%]
1.3 W (const) + 4.8 W (cap) [88%]
APU GPU
"APU GPU"
Memory
Cap
Compute
6.4 Gflop/J
100 Gflop/s [95%], 8.7 GB/s [81%]
16 W (const) + 3.2 W (cap) [100%]
GTX 580
"Desktop GPU"
Cap
Compute
Memory
5.4 Gflop/J
1.4 Tflop/s [88%], 170 GB/s [89%]
120 W (const) + 150 W (cap) [94%]
NUC CPU
"Mobile CP
Cap
Compute
3.2 Gflop/J
56 Gflop/s [97%], 18 GB/s [70%]
17 W (const) + 7.4 W (cap) [98%]
"Powerline"
Cap
Memory
Compute
2.5 Gflop/J
9.5 Gflop/s [99%], 1.3 GB/s [40%]
3.5 W (const) + 1.2 W (cap) [95%]
Cap
Memory
Compute
2.2 Gflop/J
16 Gflop/s [58%], 3.9 GB/s [31%]
5.5 W (const) + 2.0 W (cap) [97%]
APU CPU
"APU CPU"
Cap
Compute
650 Mflop/J
13 Gflop/s [98%], 3.3 GB/s [31%]
20 W (const) + 1.4 W (cap) [98%]
Desktop CPU
"Desktop CPU"
Cap
Memory
Compute
620 Mflop/J
99 Gflop/s [93%], 19 GB/s [74%]
120 W (const) + 44 W (cap) [99%]
Power [normalized]
1.15
1.1
1.05
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
1/8
1/2
2
8
32
128
512
Intensity (single−precision flop:Byte)

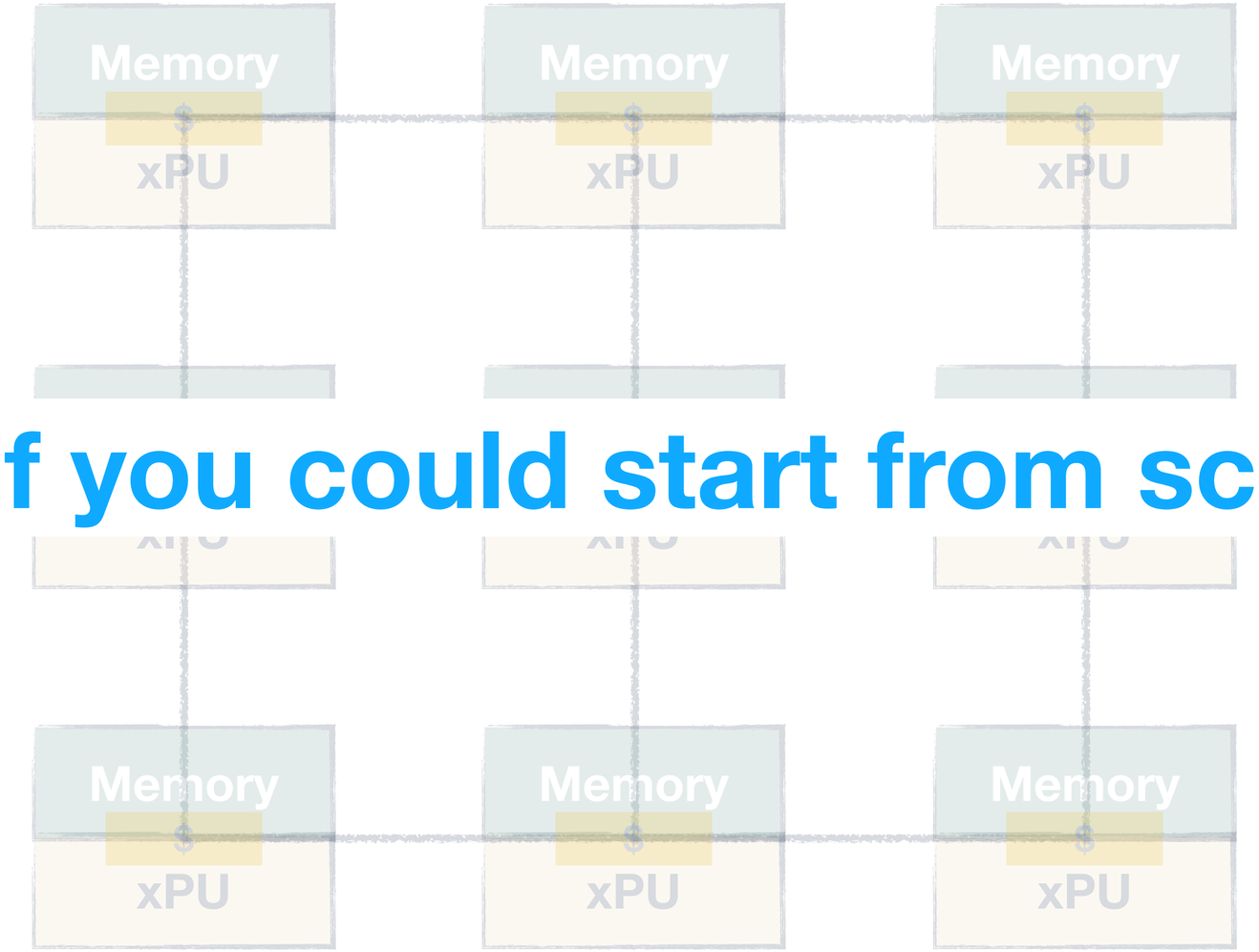


This model is good, but still too simple → **opportunity**!


GTX Titan
GTX 680
Xeon Phi
NUC GPU
Arndale GPU
APU GPU
“Desktop GPU”
“Desktop GPU”
“Desktop xPU”
“Mobile GPU”
“Mobile GPU”
“APU GPU”
GTX Titan
Arndale GPU
Power [normalized]
Cap
Memory
Compute
16 Gflop/J, 1.3 GB/J
4.0 Tflop/s [81%], 240 GB/s [83%]
8.1 Gflop/J, 1.5 GB/J
33 Gflop/s [46%], 8.4 GB/s [66%]
System designers choose
power allocations → +oppy!

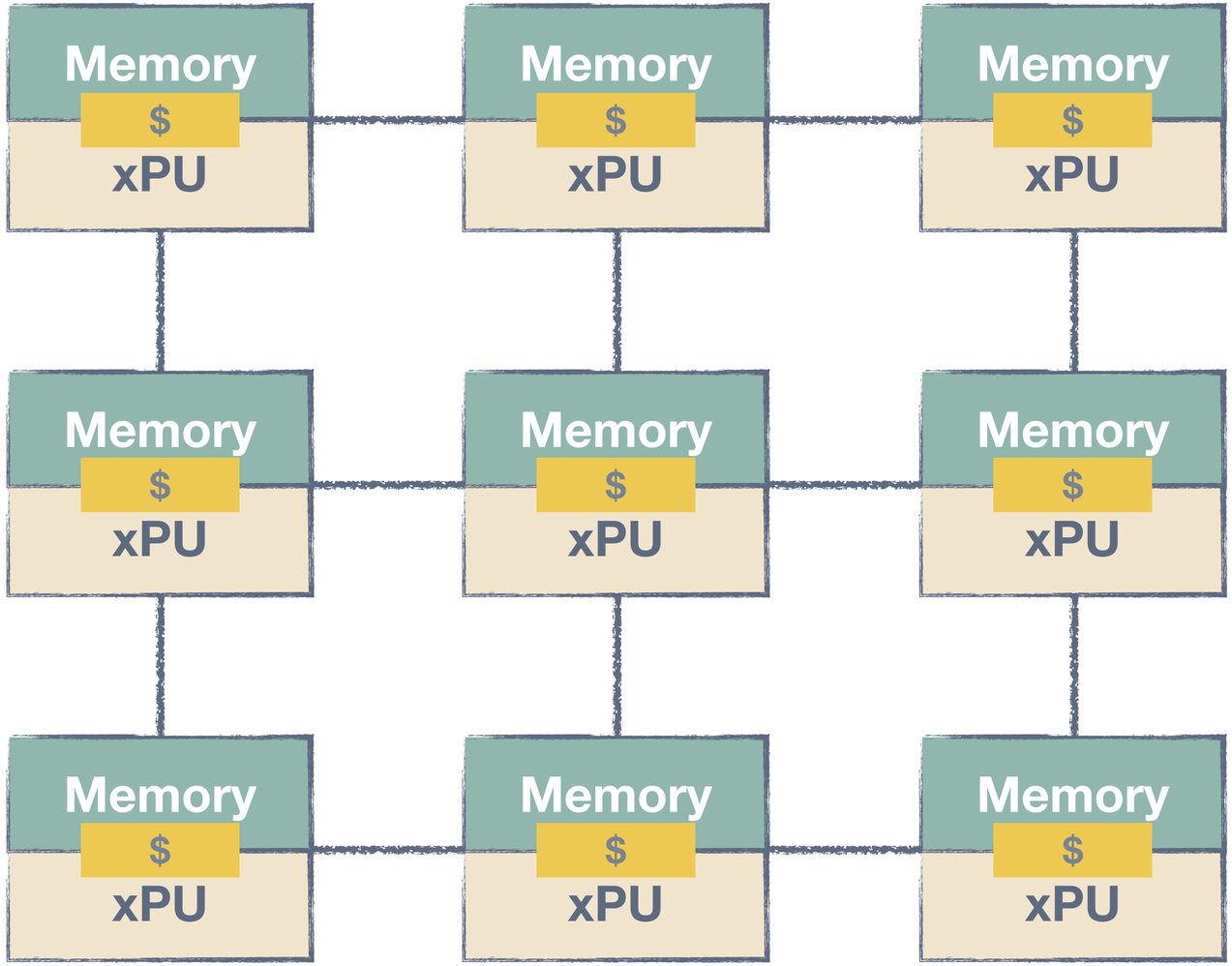
What is the “standard model?”

What can it tell me about building energy-efficient systems?



What if you could start from scratch?

K. Czechowski, R. Vuduc. *A theoretical framework for algorithm-architecture co-design*. In *IPDPS'13*. doi:10.1109/IPDPS.2013.99



K. Czechowski, R. Vuduc. *A theoretical framework for algorithm-architecture co-design*. In *IPDPS'13*. doi:10.1109/IPDPS.2013.99

Problem:

Given an algorithm

and a fixed power

& transistor budget,

pick the cores, caches, topology,

& all speeds and feeds

to minimize execution time.

K. Czechowski, R. Vuduc. *A theoretical framework for algorithm-architecture co-design*. In *IPDPS'13*. doi:10.1109/IPDPS.2013.99

Problem:

Given an algorithm

and a fixed power

& transistor budget,

pick the cores, caches, topology,

& all speeds and feeds

to minimize execution time.

(Constraints!)

K. Czechowski, R. Vuduc. *A theoretical framework for algorithm-architecture co-design*. In *IPDPS'13*. doi:10.1109/IPDPS.2013.99

Problem:

Given an algorithm
and a fixed power
& transistor budget,

pick the cores, caches, topology,
& all speeds and feeds

to minimize execution time.

K. Czechowski, R. Vuduc. *A theoretical framework for algorithm-architecture co-design*. In *IPDPS'13*. doi:10.1109/IPDPS.2013.99

Problem:

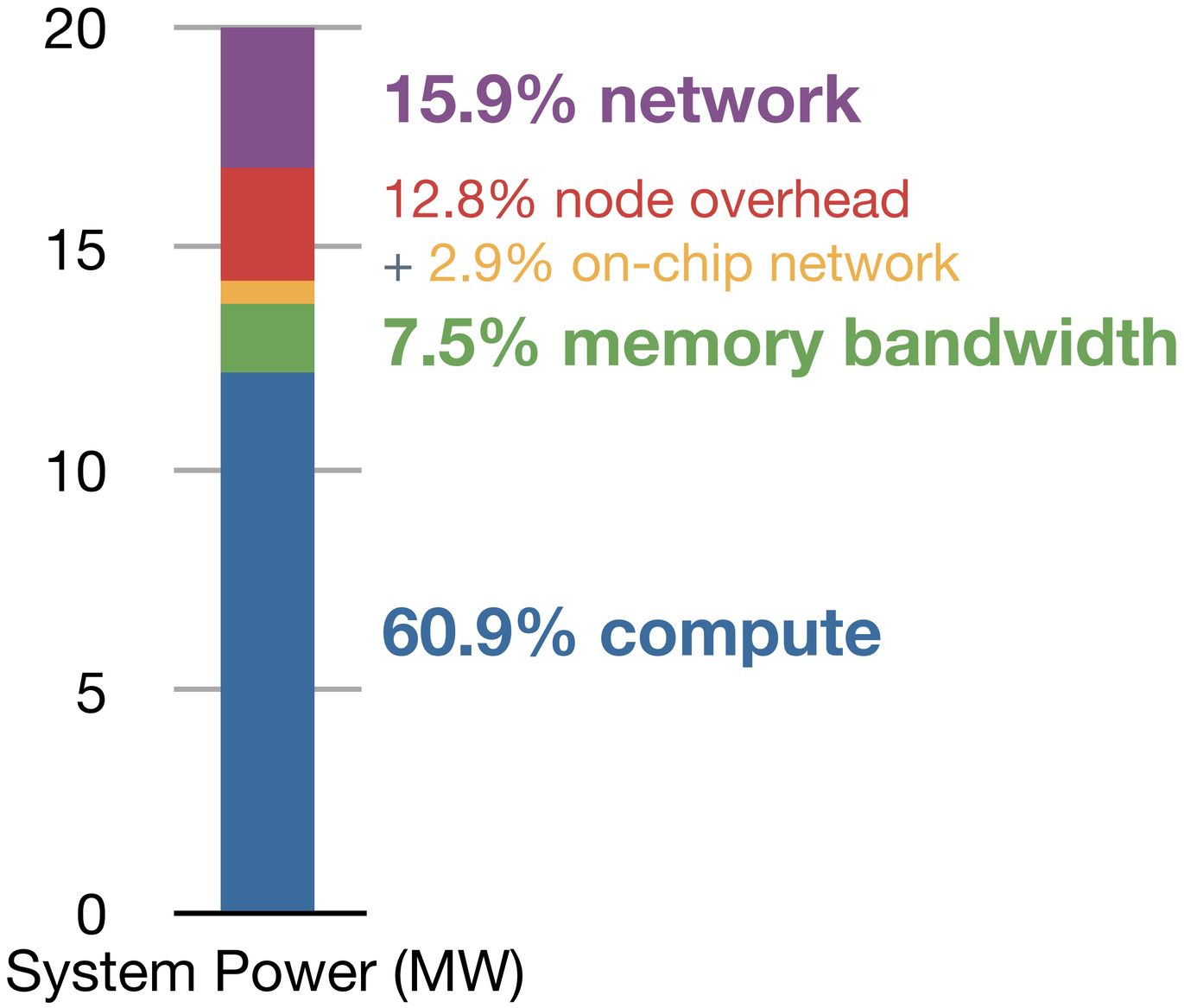
Given an algorithm
and a fixed power
& transistor budget,
pick the cores, caches, topology,
& all speeds and feeds

to minimize execution time.

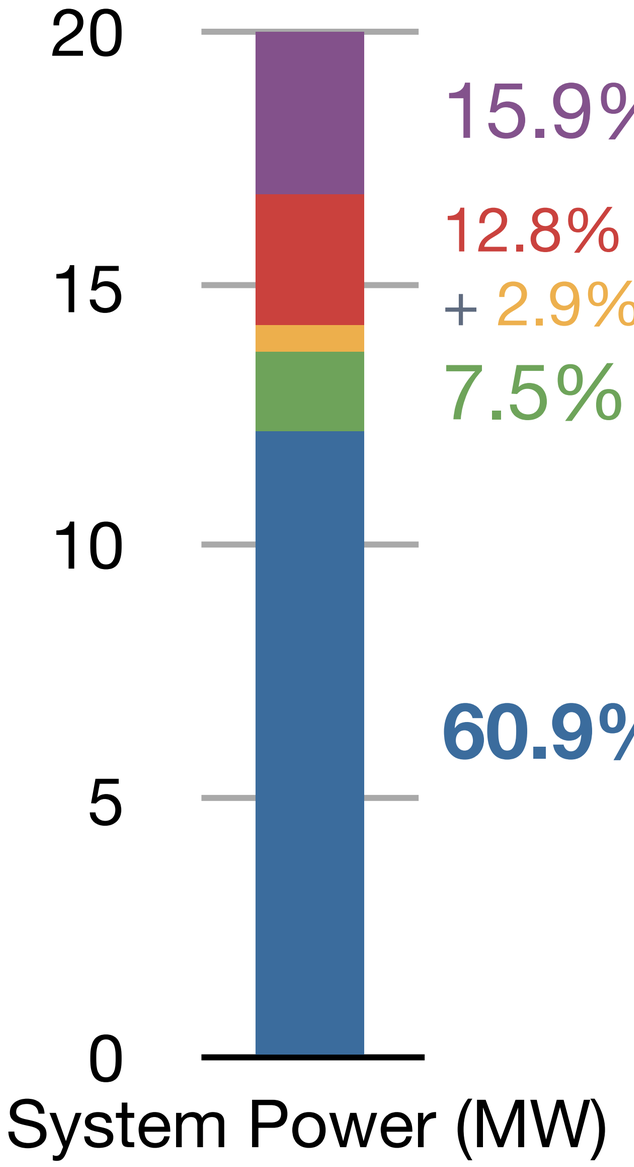
K. Czechowski, R. Vuduc. *A theoretical framework for algorithm-architecture co-design*. In *IPDPS'13*. doi:10.1109/IPDPS.2013.99

Power allocation for an “optimal” matrix multiply machine?

Power allocation for an “optimal” matrix multiply machine



Power allocation for an “optimal” matrix multiply machine



15.9% network

12.8% node overhead
+ 2.9% on-chip network

7.5% memory bandwidth

60.9% compute

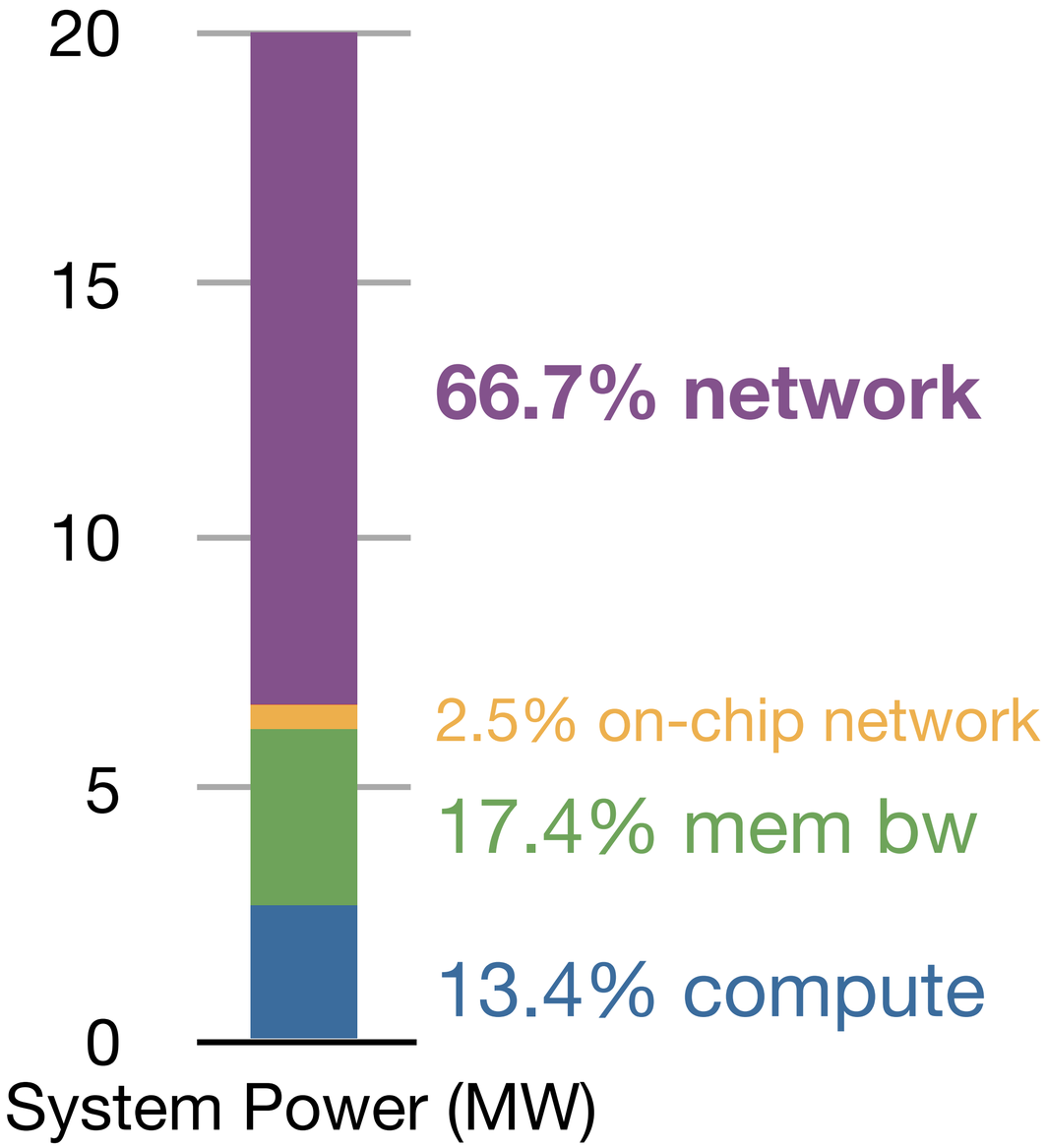
ORNL Summit (13-14 MW):
67.0% GPU compute
14.9% CPU compute

4.8% memory
5.3% network + disk
8% node overhead

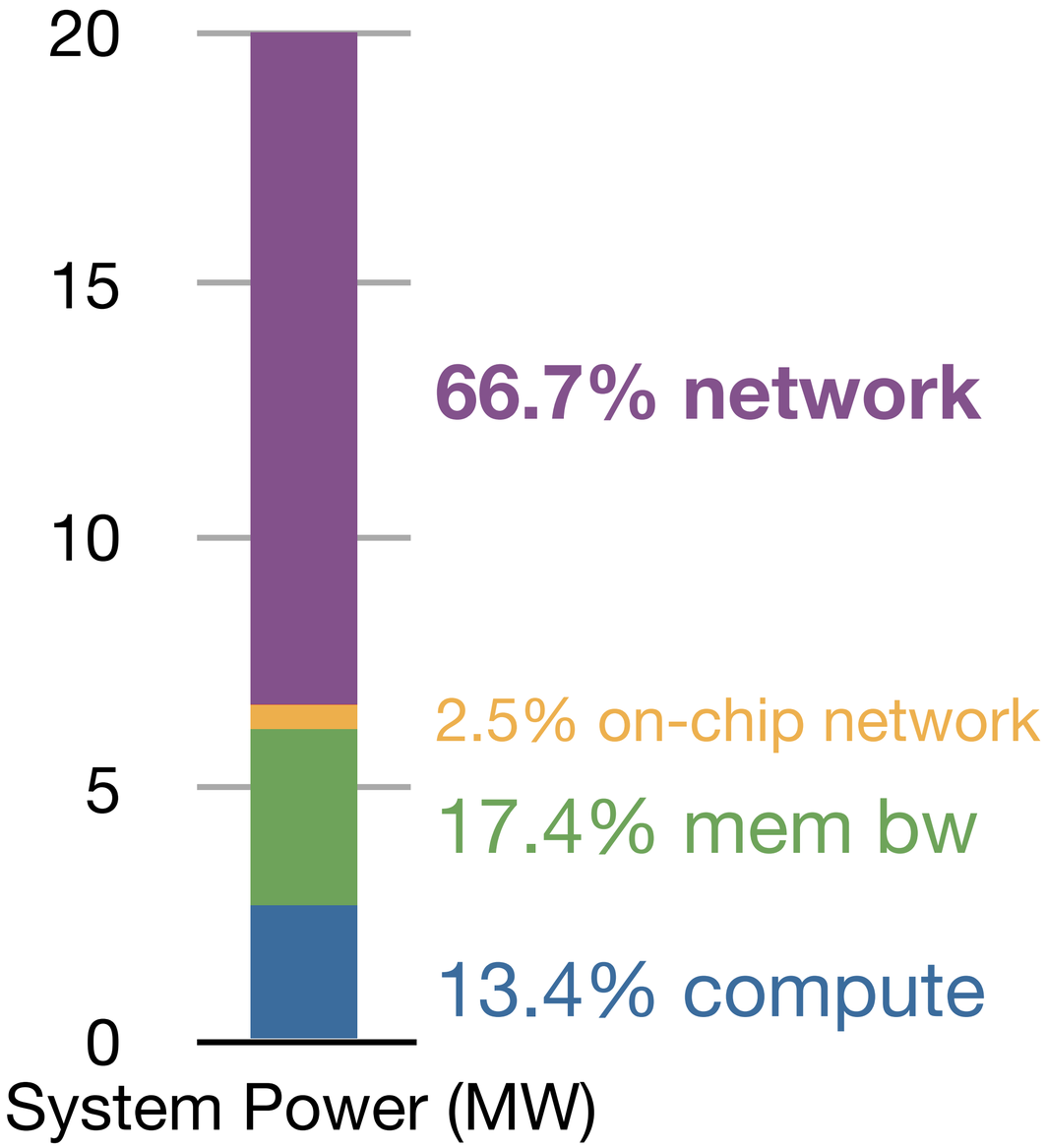
P.S.: R_{max} / R_{peak} ~ 75%

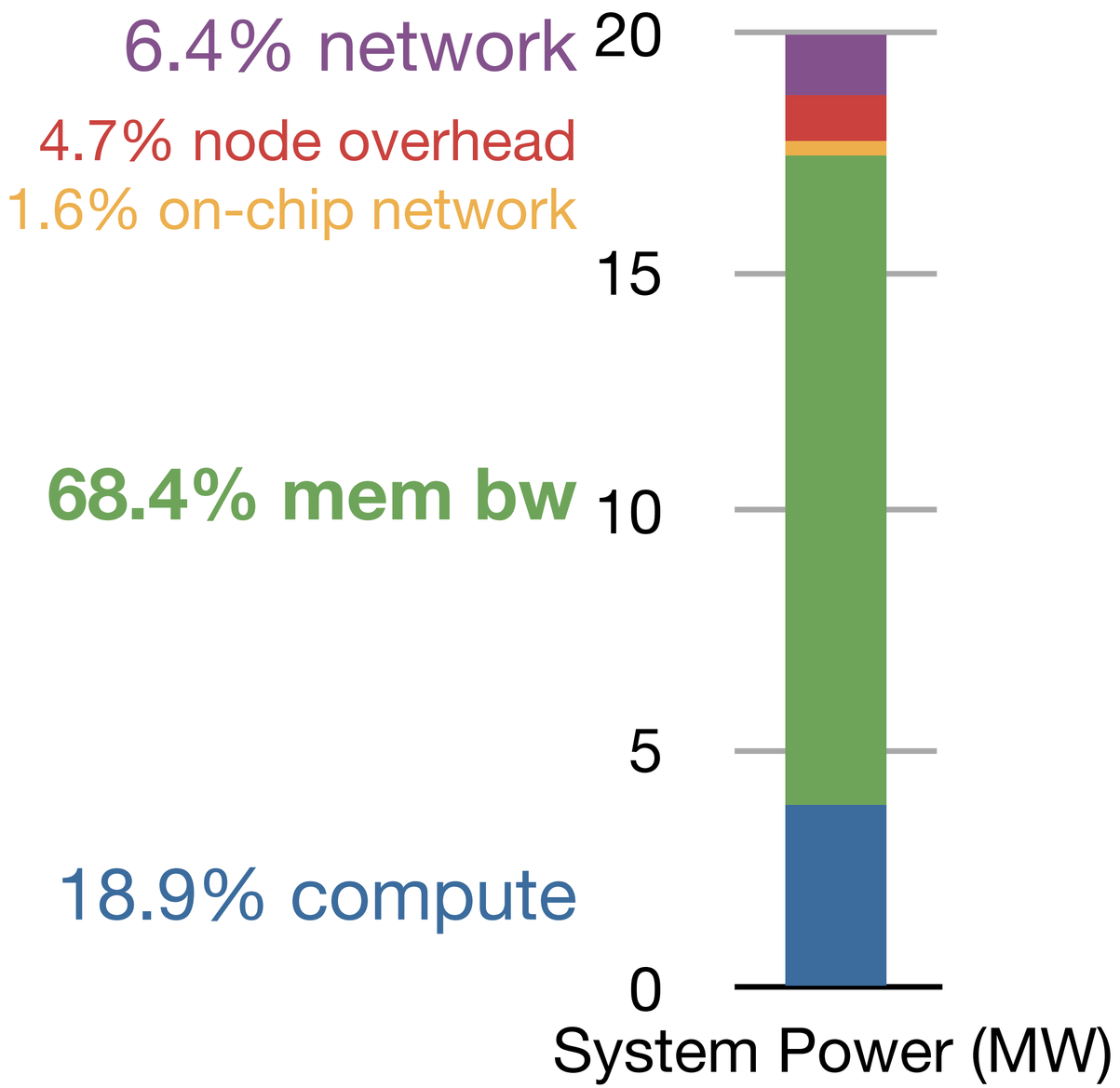
Power allocation for an “optimal” 3D FFT machine?

Power allocation for an “optimal” 3D FFT machine



3D FFT vs. “Stencil” machines



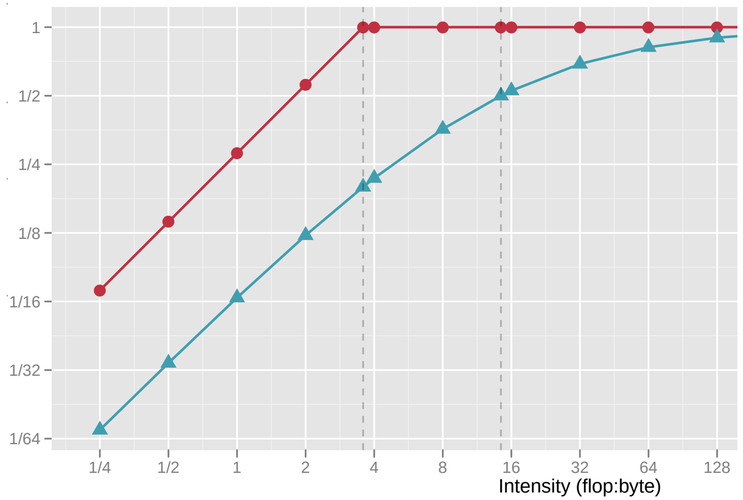


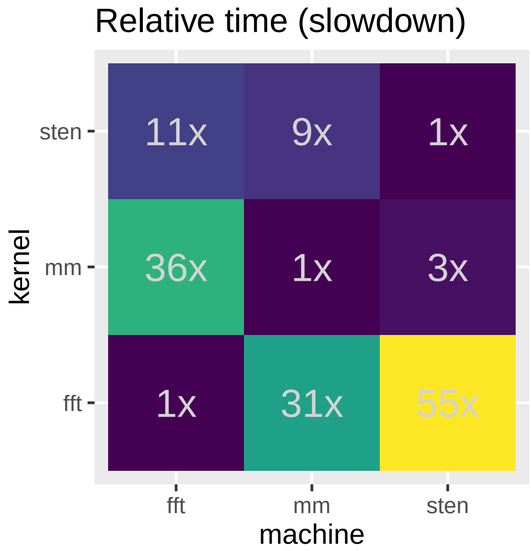

Relative time (slowdown)
kernel
sten
mm
fft
fft
mm
sten
machine

Can you beat the standard model?

- **What is your computational model?**
- Can I teach it to a freshman? (“Conte” question)
- Is it productive?
- Is it “easy” to provide provably guarantees for your model?
- **Can your model easily accommodate a variety of physical costs?**

These slides: bit.ly/4elibtM





What about LLM codesign, you ask? For a partial (non-energy) answer, see our SC23 paper on **Calculon** *by M. Isaev et al.:* doi:10.1145/3581784.3607102