These slides + links to papers: hpcgarage.org/salishan23

Embracing communication ~~and unreliability~~

SALISHAN 2023

RICH VUDUC – APRIL 26

Georgia Tech College of Computing
School of Computational Science and Engineering

Georgia Tech College of Computing
Center for Research into Novel Computing Hierarchies

Rank	Name	Accel	Rmax:PF/s	HPCG:PF/s	GF/J	GB/J (est.)
8	Perlmutter	NVIDIA A100 SXM4 40 GB	70.9	1.9	27.4	4.4
1	Frontier	AMD Instinct MI250X	1102.0	14.1	52.2	4.0
9	Selene	NVIDIA A100	63.5	1.6	24.0	3.7
11	Adastra	AMD Instinct MI250X	46.1	0.6	50.0	3.7
3	LUMI	AMD Instinct MI250X	309.1	3.4	51.4	3.4
2	Supercomputer Fugaku	None	442.0	16.0	14.8	3.2
4	Leonardo	NVIDIA A100 SXM4 64 GB	174.7	2.6	31.1	2.7
5	Summit	NVIDIA Volta GV100	148.6	2.9	14.7	1.7
6	Sierra	NVIDIA Volta GV100	94.6	1.8	12.7	1.4
7	Sunway TaihuLight	None	93.0	0.5	6.1	0.2

Rank	Name	Accel	Rmax:PF/s	HPCG:PF/s	GF/J	GB/J (est.)
8	Perlmutter	NVIDIA A100 SXM4 40 GB	70.9	1.9	27.4	4.4
1	Frontier	AMD Instinct MI250X	1102.0	14.1	52.2	4.0
9	Selene	NVIDIA A100	63.5	1.6	24.0	3.7
11	Adastra	AMD Instinct MI250X	46.1	0.6	50.0	3.7
3	LUMI	AMD Instinct MI250X	309.1	3.4	51.4	3.4
2	Supercomputer Fugaku	None	442.0	16.0	14.8	3.2
4	Leonardo	NVIDIA A100 SXM4 64 GB	174.7	2.6	31.1	2.7
5	Summit	NVIDIA Volta GV100	148.6	2.9	14.7	1.7
6	Sierra	NVIDIA Volta GV100	94.6	1.8	12.7	1.4
7	Sunway TaihuLight	None	93.0	0.5	6.1	0.2

Rank	Name	Accel	Rmax:PF/s	HPCG:PF/s	GF/J	GB/J (est.)
8	Perlmutter	NVIDIA A100 SXM4 40 GB	70.9	1.9	27.4	4.4
1	Frontier	AMD Instinct MI250X	1102.0	14.1	52.2	4.0
9	Selene	NVIDIA A100	63.5	1.6	24.0	3.7
11	Adastra	AMD Instinct MI250X	46.1	0.6	50.0	3.7
3	LUMI	AMD Instinct MI250X	309.1	3.4	51.4	3.4
2	Supercomputer Fugaku	None	442.0	16.0	14.8	3.2
4	Leonardo	NVIDIA A100 SXM4 64 GB	174.7	2.6	31.1	2.7
5	Summit	NVIDIA Volta GV100	148.6	2.9	14.7	1.7
6	Sierra	NVIDIA Volta GV100	94.6	1.8	12.7	1.4
7	Sunway TaihuLight	None	93.0	0.5	6.1	0.2

Four "generations" of computing

Gregory Abowd (2016). "Beyond Weiser: From ubiquitous computing to collective computing." DOI: 10.1109/MC.2016.22

OUTLOOK

TABLE 1. A framework for comparing computing generations, inspired by Mark Weiser.

Generation	Time frame	Human–computer ratio	Canonical device	Application	
				Initial	Follow-on
1	Mid-1930s	Many–1	Mainframe	Scientific calculation	Data processing
2	Late 1960s	1–1	PC	Spreadsheet	Database management, document processing
3	Late 1980s	1–many	Inch/foot/yard	Calendar and contact management, human–human communication	Location-based services, social media, app ecosystem, education
4	Mid-2000s	Many–many	Cloud/crowd/shroud	Personal navigation and entertainment	Health advisors, educational assistants, supply chain logistics

Four "generations" of computing

Gregory Abowd (2016). "Beyond Weiser: From ubiquitous computing to collective computing." DOI: 10.1109/MC.2016.22

OUTLOOK

TABLE 1. A framework for comparing computing generations, inspired by Mark Weiser.

Generation	Time frame	Human–computer ratio	Canonical device	Application: Initial	Application: Follow-on
1	Mid-1930s	Many–1	Mainframe	Scientific calculation	Data processing
2	Late 1960s	1–1	PC	Spreadsheet	Database management, document processing
3	Late 1980s	1–many	Inch/foot/yard	Calendar and contact management, human–human communication	Location-based services, social media, app ecosystem, education
4	Mid-2000s	Many–many	Cloud/crowd/shroud	Personal navigation and entertainment	Health advisors, educational assistants, supply chain logistics

Four “generations” of computing

Gregory Abowd (2016). “Beyond Weiser: From ubiquitous computing to collective computing.” DOI: 10.1109/MC.2016.22

OUTLOOK

TABLE 1. A framework for comparing computing generations, inspired by Mark Weiser.

Generation	Time frame	Human–computer ratio	Canonical device	Application	
				Initial	Follow-on
1	Mid-1930s	Many–1	Mainframe	Scientific calculation	Data processing
2	Late 1960s	1–1	PC	Spreadsheet	Database management, document processing
3	Late 1980s	1–many	Inch/foot/yard	Calendar and contact management, human–human communication	Location-based services, social media, app ecosystem, education
4	Mid-2000s	Many–many	Cloud/crowd/shroud	Personal navigation and entertainment	Health advisors, educational assistants, supply chain logistics

Session title:

Q: **Is the cloud for everyone**, e.g., HPC people?

Session title:

Q: **Is the cloud for everyone**, e.g., HPC people?

A: Sure, why not?

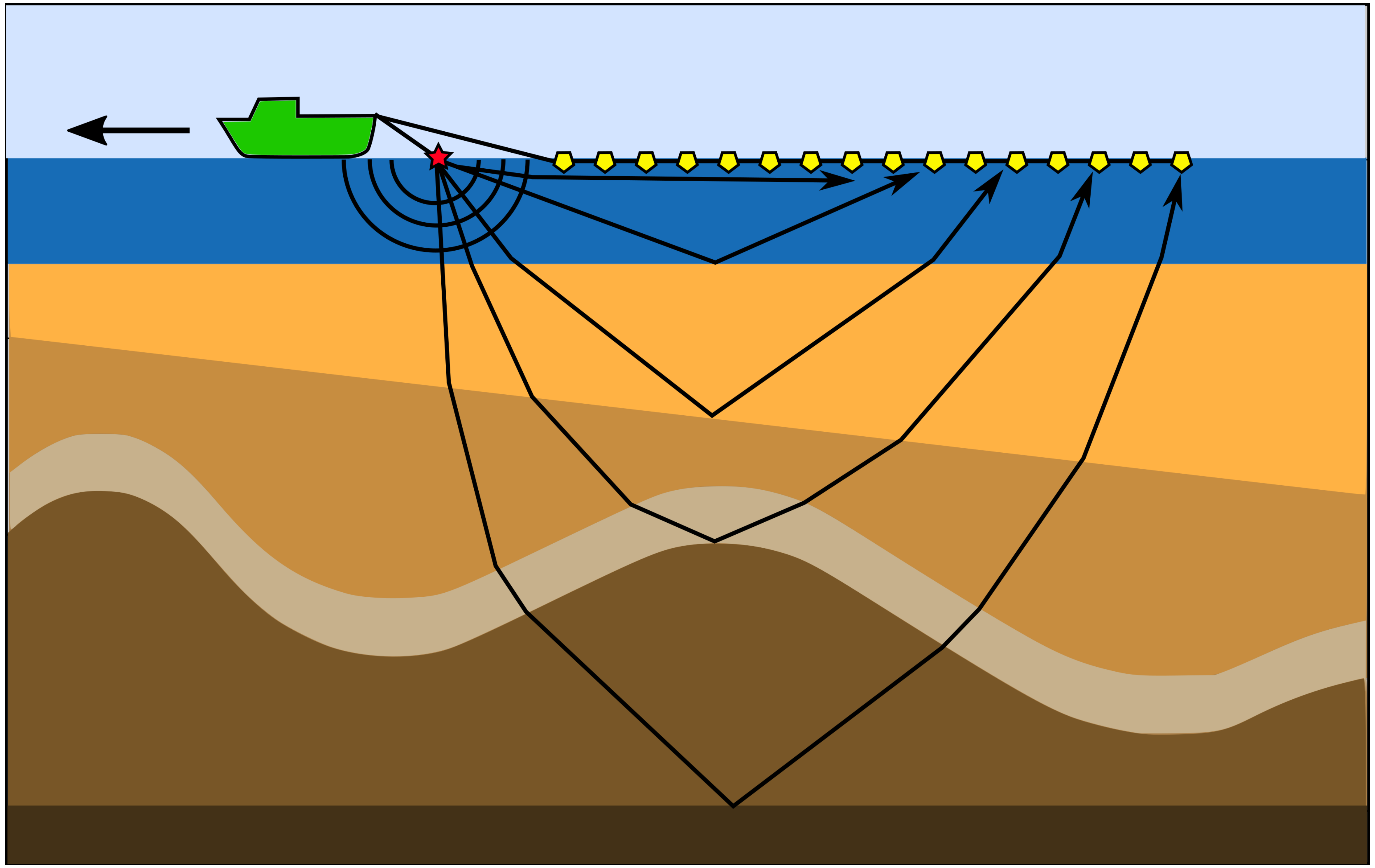
Myth 12: All HPC Will Be Subsumed by the Clouds!

The rapidly advancing AI and new precision options has reignited the cloud discussion. The question whether clouds will subsume supercomputing has been ongoing for more than a decade, since the late 2000s Deelman et al. (2008), but remains inconclusive. Today's cloud offerings offer a wide spectrum for HPC customers, ranging from low-cost standard virtual machines to specialized top-gear HPC equipment in

S. Matsuoka (2023). *"Myths and legends in high-performance computing." arXiv:* 2301.02432

Serverless seismic inversion (2018-2019)

– **Philipp Witte** (GT Ph.D., now @ **MSR**), Felix Herrmann (advisor)



P. Witte (2020). *"Software and algorithms for large-scale seismic inverse problems."* Ph.D. Dissertation at GT. https://hdl.handle.net/1853/62754

Serverless seismic inversion (2018-2019)

– **Philipp Witte** (GT Ph.D., now @ **MSR**), Felix Herrmann (advisor)

$$\underset{\mathbf{m}}{\text{minimize}} \;\; \Phi(\mathbf{m}) = \sum_{i=1}^{n_s} \frac{1}{2} ||\mathcal{F}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i||_2^2,$$

P. Witte (2020). *"Software and algorithms for large-scale seismic inverse problems."* Ph.D. Dissertation at GT. https://hdl.handle.net/1853/62754

Serverless seismic inversion (2018-2019)

– **Philipp Witte** (GT Ph.D., now @ **MSR**), Felix Herrmann (advisor)

$$\underset{\mathbf{m}}{\text{minimize}} \;\; \Phi(\mathbf{m}) = \sum_{i=1}^{n_s} \frac{1}{2} ||\mathcal{F}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i||_2^2,$$

Data

P. Witte (2020). *"Software and algorithms for large-scale seismic inverse problems."* Ph.D. Dissertation at GT. https://hdl.handle.net/1853/62754

Serverless seismic inversion (2018-2019)

– **Philipp Witte** (GT Ph.D., now @ **MSR**), Felix Herrmann (advisor)

$$\underset{\mathbf{m}}{\text{minimize}} \;\; \Phi(\mathbf{m}) = \sum_{i=1}^{n_s} \frac{1}{2} ||\mathcal{F}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i||_2^2,$$

Data

Parameters

P. Witte (2020). *"Software and algorithms for large-scale seismic inverse problems."* Ph.D. Dissertation at GT. https://hdl.handle.net/1853/62754

Serverless seismic inversion (2018-2019)

– **Philipp Witte** (GT Ph.D., now @ **MSR**), Felix Herrmann (advisor)

PDE solver

$$\underset{\mathbf{m}}{\text{minimize}} \;\; \Phi(\mathbf{m}) = \sum_{i=1}^{n_s} \frac{1}{2} ||\mathcal{F}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i||_2^2,$$

Data

Parameters

P. Witte (2020). *"Software and algorithms for large-scale seismic inverse problems."* Ph.D. Dissertation at GT. https://hdl.handle.net/1853/62754

Serverless seismic inversion (2018-2019)

– **Philipp Witte** (GT Ph.D., now @ **MSR**), Felix Herrmann (advisor)

$$\underset{\mathbf{m}}{\text{minimize}} \;\; \Phi(\mathbf{m}) = \sum_{i=1}^{n_s} \frac{1}{2} || \mathcal{F}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i ||_2^2,$$

Gradients: $$\mathbf{g} = \sum_{i=1}^{n_s} \mathbf{J}^\top \Big(\mathcal{F}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i \Big),$$

P. Witte (2020). *"Software and algorithms for large-scale seismic inverse problems."* Ph.D. Dissertation at GT. https://hdl.handle.net/1853/62754

Serverless seismic inversion (2018-2019)

– **Philipp Witte** (GT Ph.D., now @ **MSR**), Felix Herrmann (advisor)

$$\underset{\mathbf{m}}{\text{minimize}} \;\; \Phi(\mathbf{m}) = \sum_{i=1}^{n_s} \frac{1}{2} ||\mathcal{F}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i||_2^2,$$

Gradients: $$\mathbf{g} = \sum_{i=1}^{n_s} \mathbf{J}^\top \Big(\mathcal{F}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i \Big),$$

Run a PDE solver

P. Witte (2020). *"Software and algorithms for large-scale seismic inverse problems."* Ph.D. Dissertation at GT. https://hdl.handle.net/1853/62754

Serverless seismic inversion (2018-2019)

– **Philipp Witte** (GT Ph.D., now @ **MSR**), Felix Herrmann (advisor)

Run for all *i* independently

("Map", AWS/Azure Batch → S3/Blob)

Gradients: $$\mathbf{g} = \sum_{i=1}^{n_s} \boxed{\mathbf{J}^\top \Big(\mathcal{F}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i \Big)},$$

P. Witte (2020). *"Software and algorithms for large-scale seismic inverse problems."* Ph.D. Dissertation at GT. https://hdl.handle.net/1853/62754

Serverless seismic inversion (2018-2019)

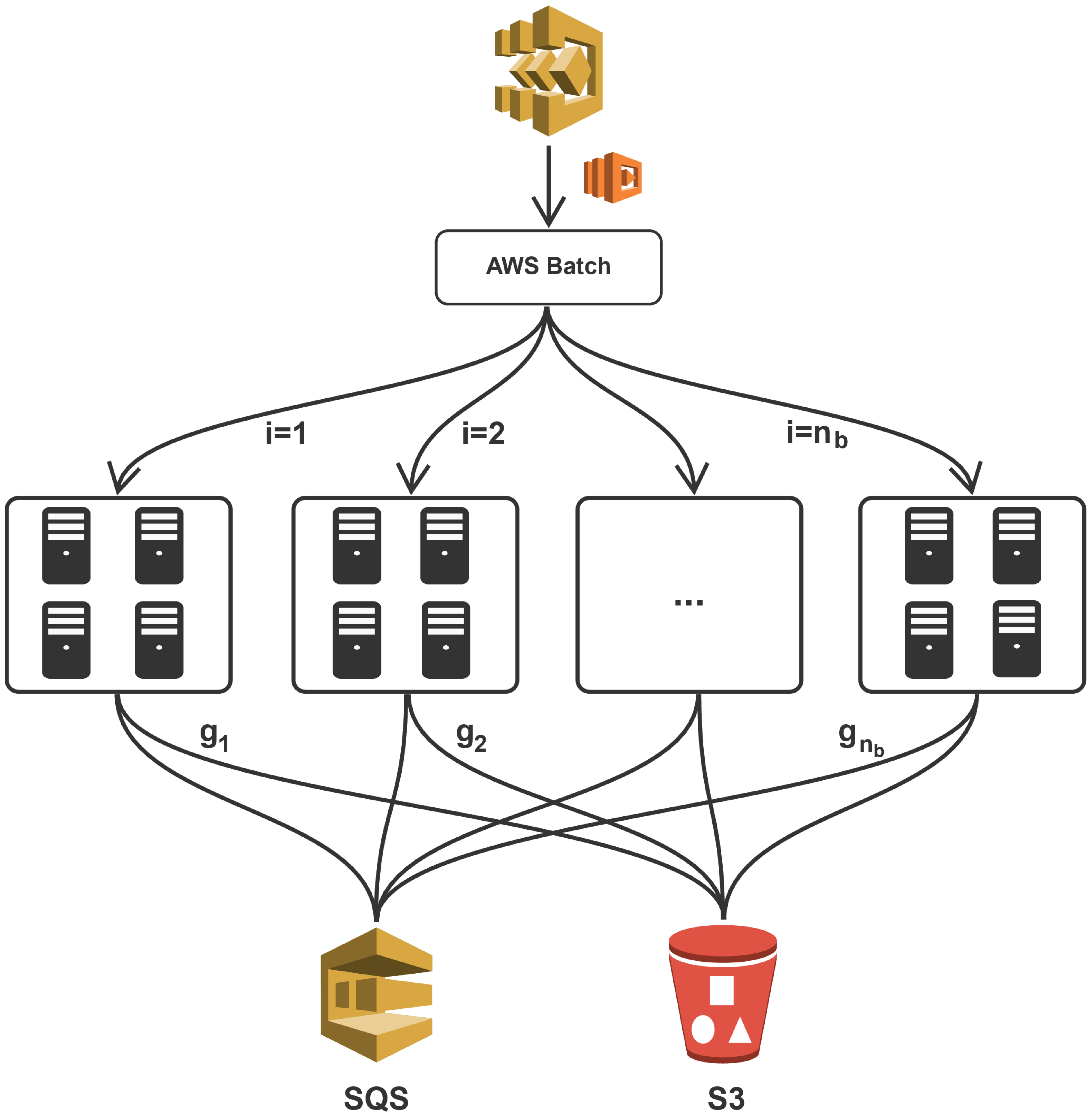
– **Philipp Witte** (GT Ph.D., now @ **MSR**), Felix Herrmann (advisor)

Gradients: $$\mathbf{g} = \boxed{\sum_{i=1}^{n_s}} \mathbf{J}^\top \Big(\mathcal{F}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i \Big),$$

Event-driven

(AWS Lambda+SQS / Azure equivs.)

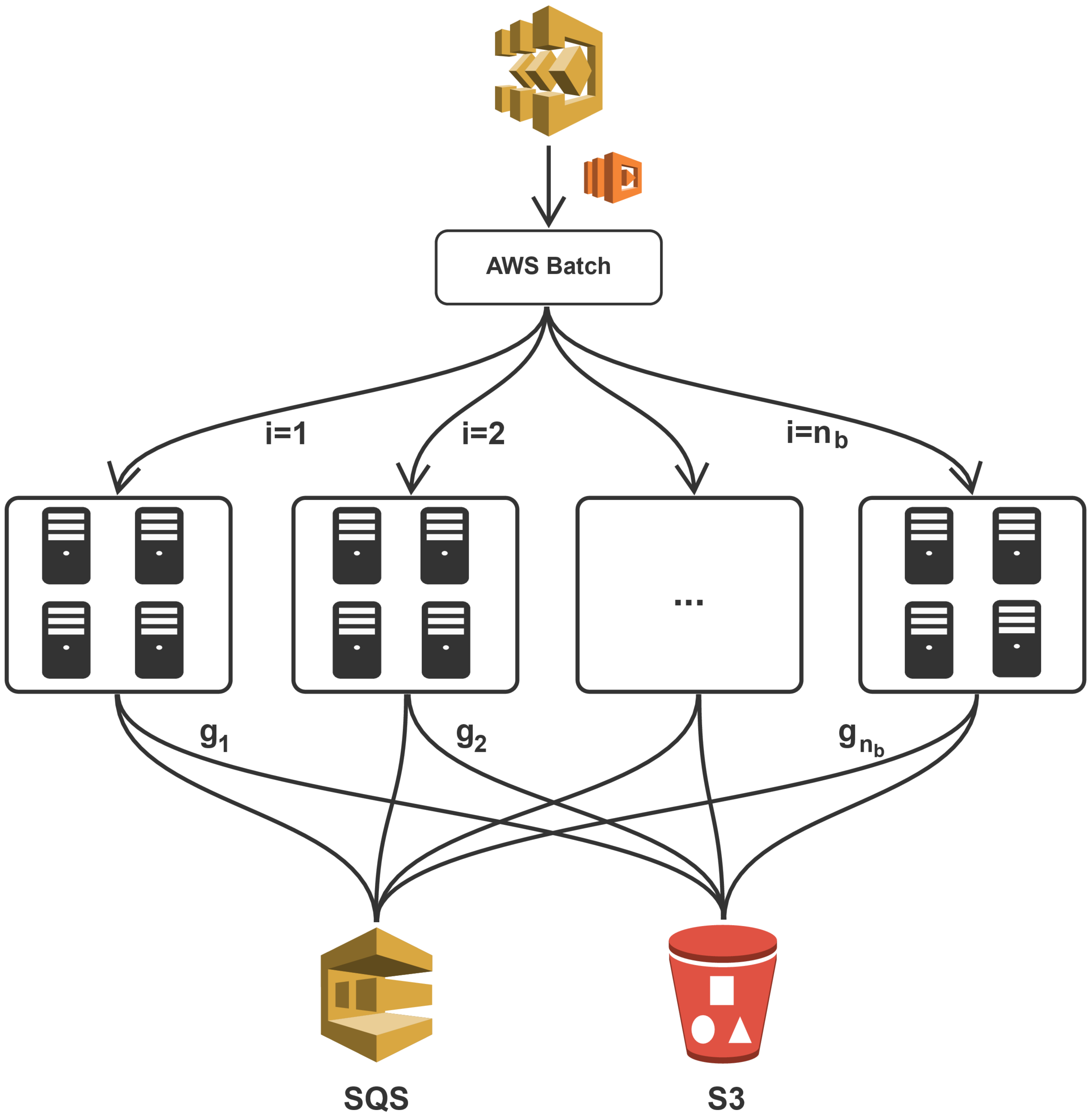
P. Witte (2020). *"Software and algorithms for large-scale seismic inverse problems."*

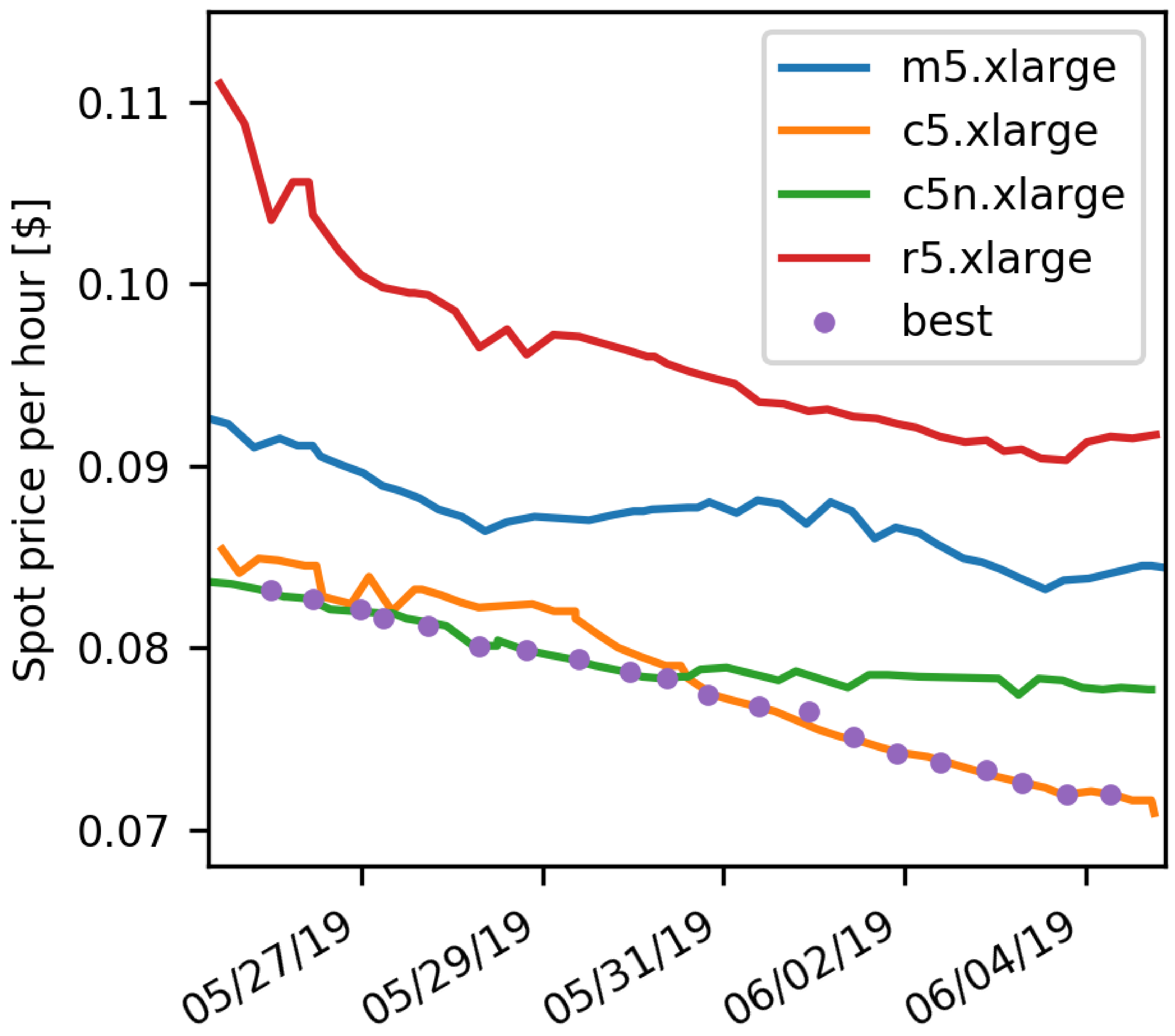


Gradients:

$$\mathbf{g} = \sum_{i=1}^{n_s} \mathbf{J}^\top \Big(\mathcal{F}(\mathbf{m}, \mathbf{q}_i) - \mathbf{d}_i \Big),$$

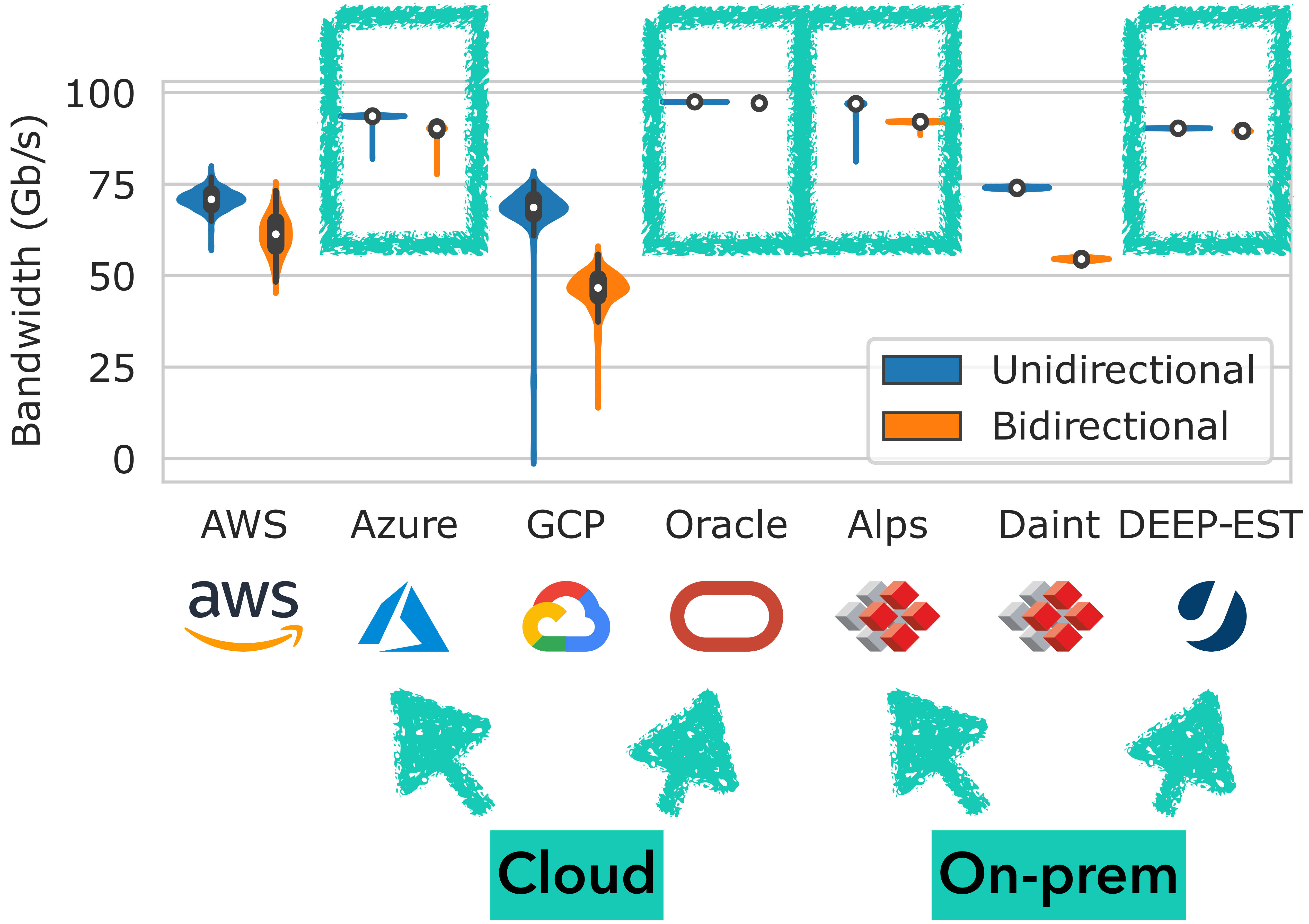
P. Witte (2020). *"Software and algorithms for large-scale seismic inverse problems."* Ph.D. Dissertation at GT. https://hdl.handle.net/1853/62754



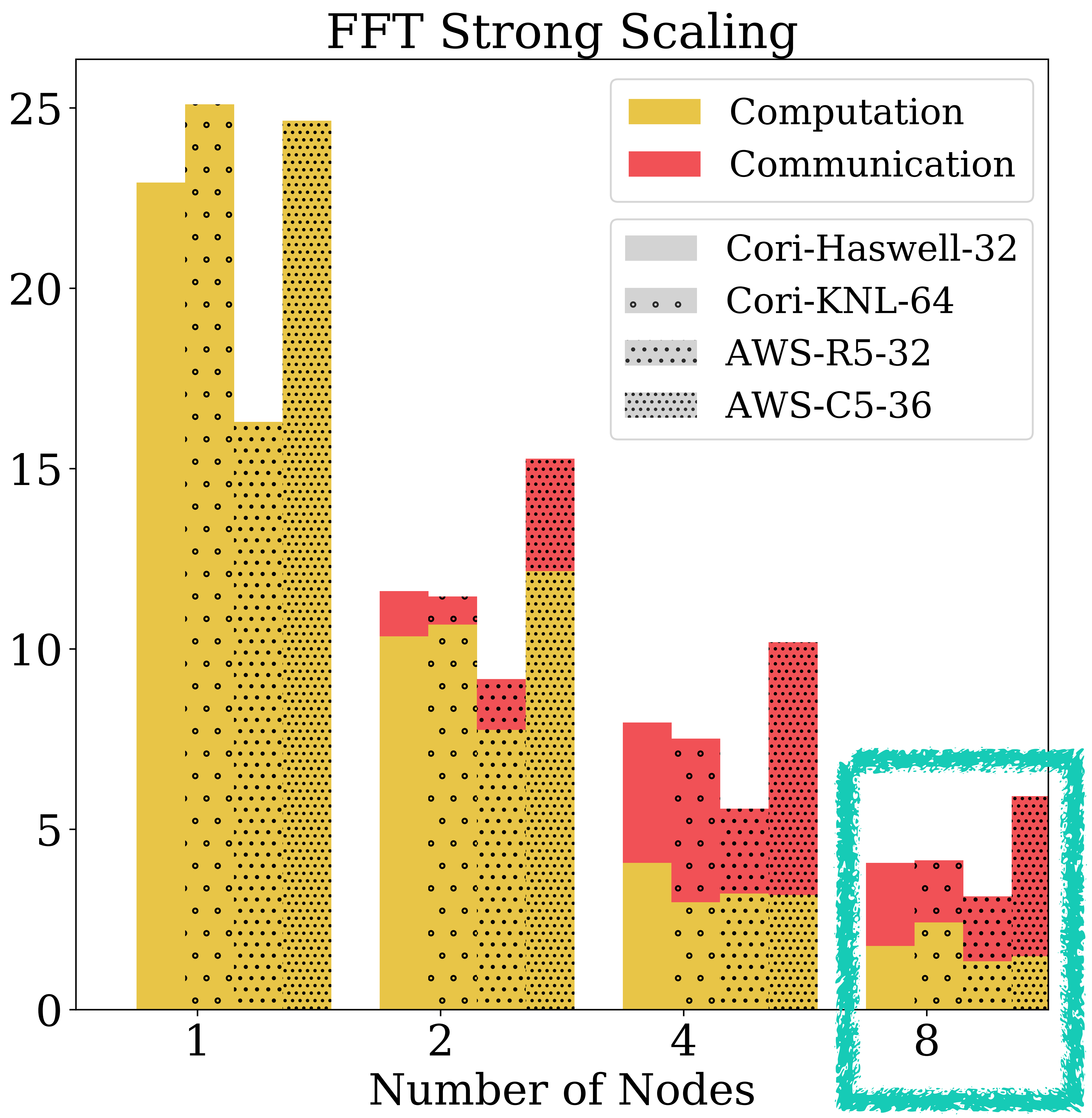


(a)

P. Witte (2020). *"Software and algorithms for large-scale seismic inverse problems."* Ph.D. Dissertation at GT. https://hdl.handle.net/1853/62754



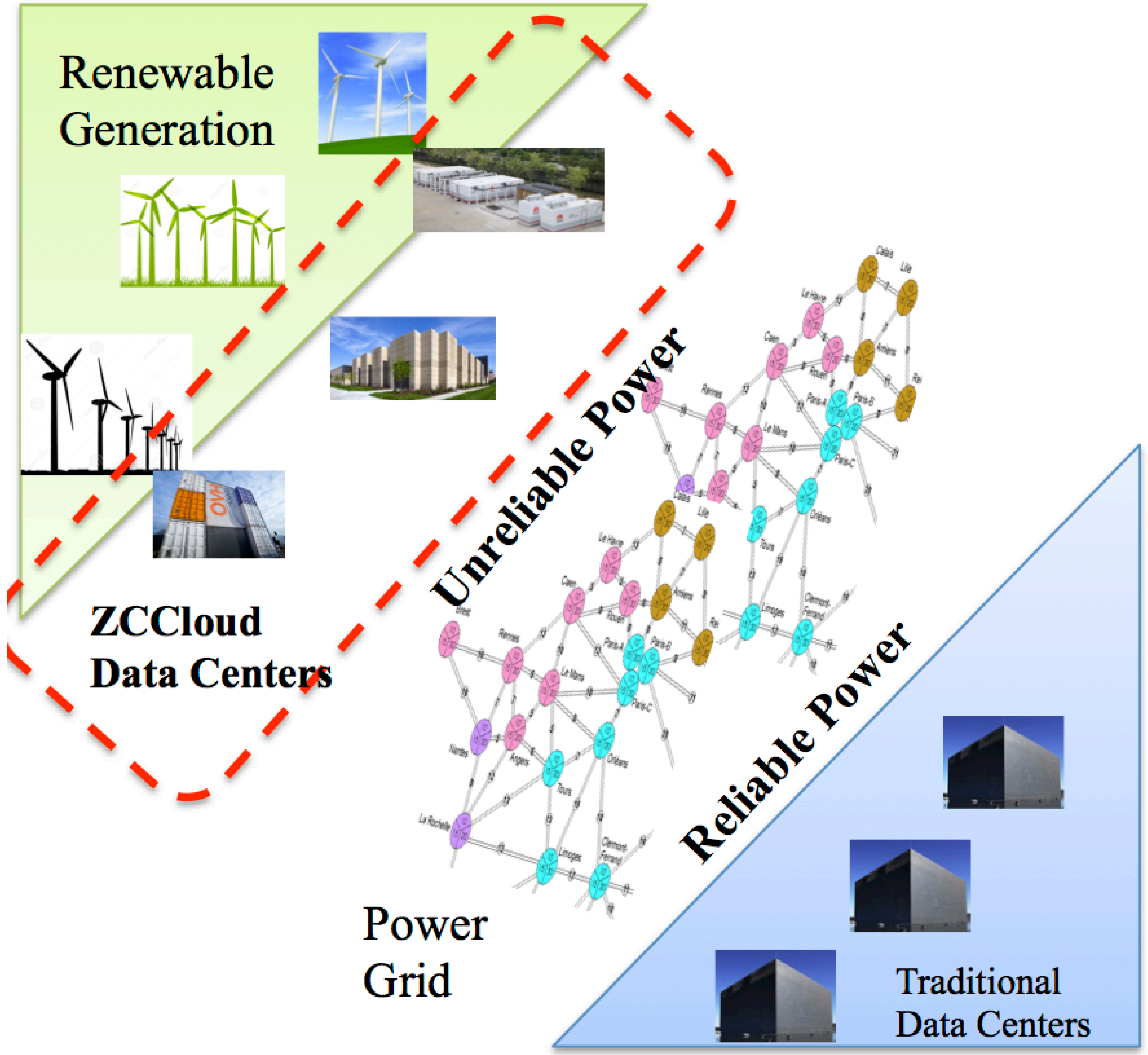
Observation 6: Latency noise affects both cloud and on-premise HPC systems, and can increase the latency by more than 100x (in a single case up to 35000x). Except that for GCP, HPC instances are not characterized by a lower latency noise than normal instances.



De Sensi et al. (2022). "Noise in the clouds: influence of network performance variability on application scalability." arXiv:2210.15315

Guidi et al. (2021). "Ten years later: cloud computing is closing the performance gap."

The Future Cloud: Traditional+ZCCloud



- Volatile Resources running on Stranded Power complement Traditional
- Computing load shaped to be "compliant"
  - Time shifting
  - Location-shifting
- All that can be is shifted
  - Economic
  - Environmental

Chien et al. (2015). "The Zero-Carbon Cloud: High-value, dispatch able demand for renewable power generators." doi:10.1016/j.tej.2015.09.010

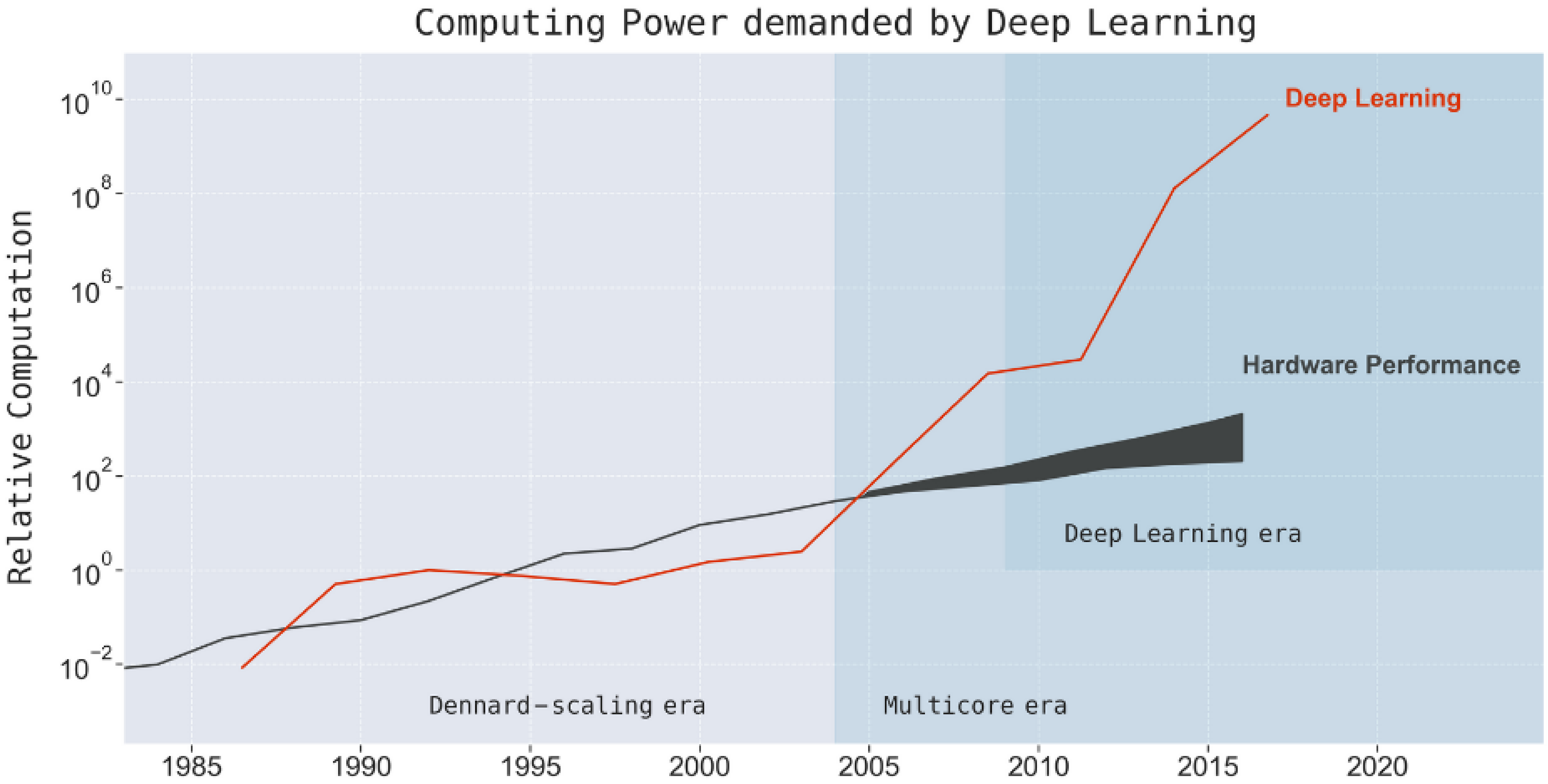
Q: Is the cloud for everyone, e.g., HPC people?

A: Sure, why not?

Q: **What system will the cloud provide?**

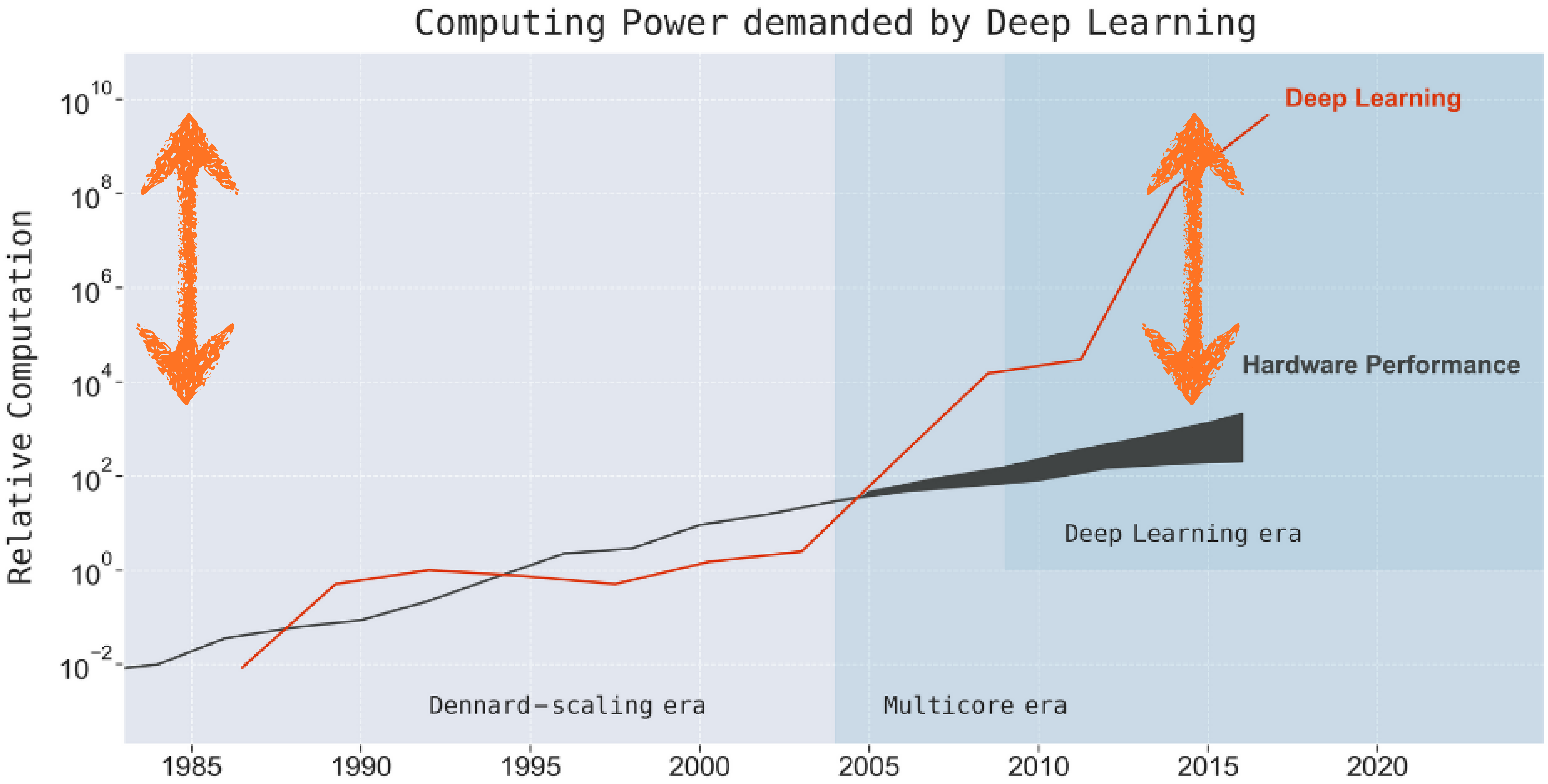
Satiating the beast…

The demand for compute due to deep learning far outstrips the what Moore's Law-like hardware performance can deliver. Source: **Thompson et al. (2020)**: "The computational limits of deep learning." arXiv:2007.05558v1



Satiating the beast…

The demand for compute due to deep learning far outstrips the what Moore's Law-like hardware performance can deliver. Source: **Thompson et al. (2020)**: "The computational limits of deep learning." arXiv:2007.05558v1



(a) Hardware Architectures

(b) Training with Single vs. Multiple Nodes

Fig. 3. Parallel Architectures in Deep Learning

Ben-Nun & Hoefler (2019). "Demystifying parallel and distributed deep learning: an in-depth concurrency analysis."

doi:10.1145/3320060

Q: Is the cloud for everyone, e.g., HPC people?
A: Sure, why not?

Q: What system will the cloud provide?
Q: What is an optimal GPU machine for DL?

Canonical structure of a large language model

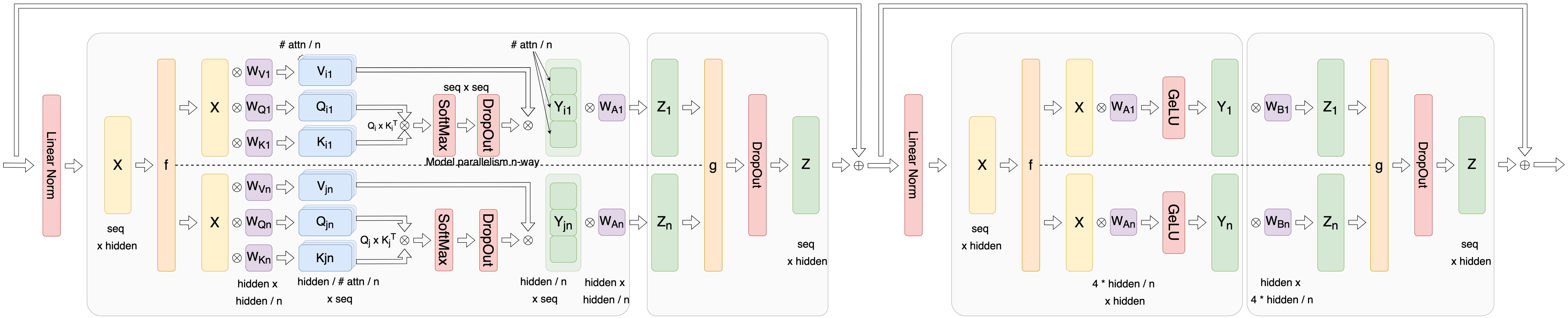
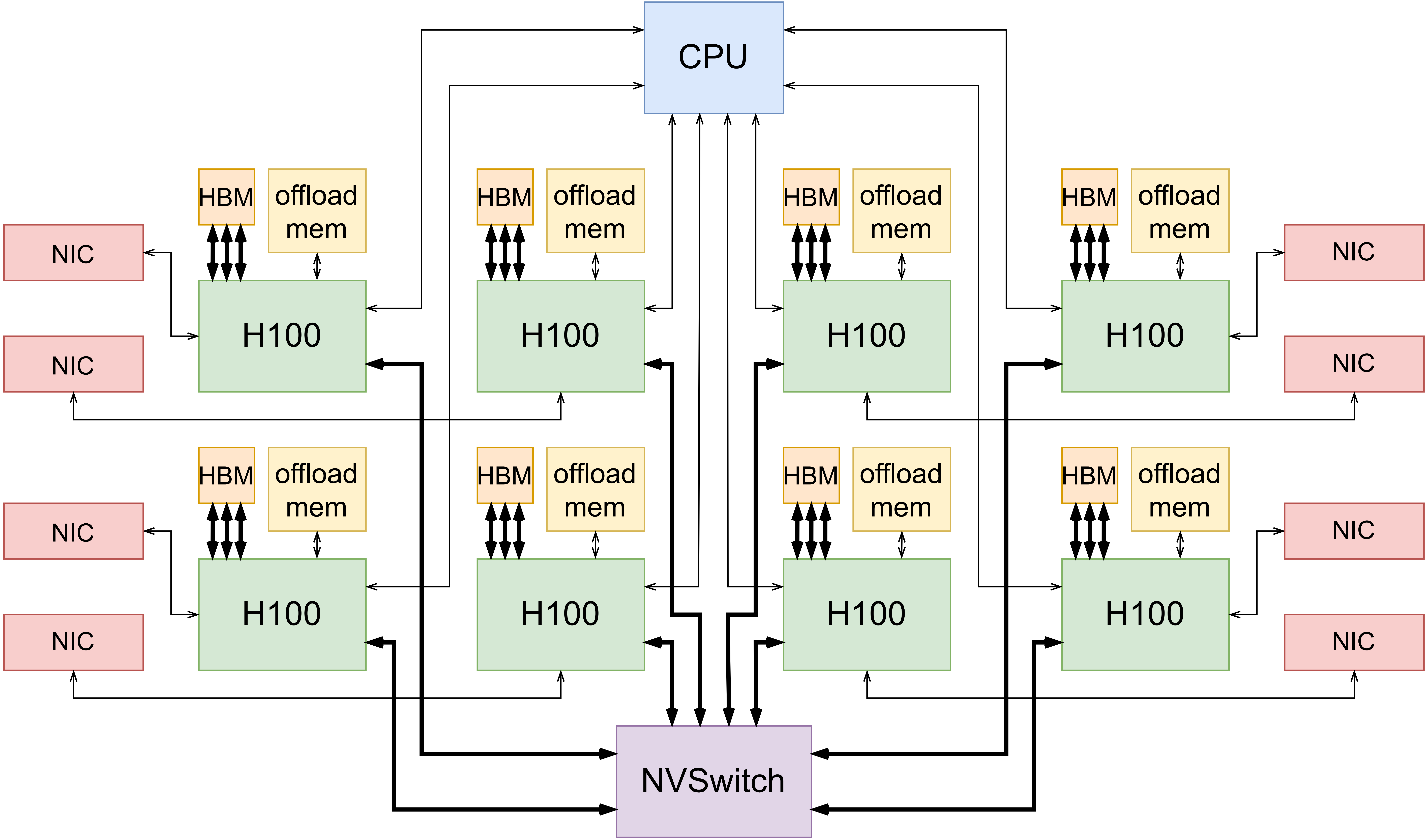
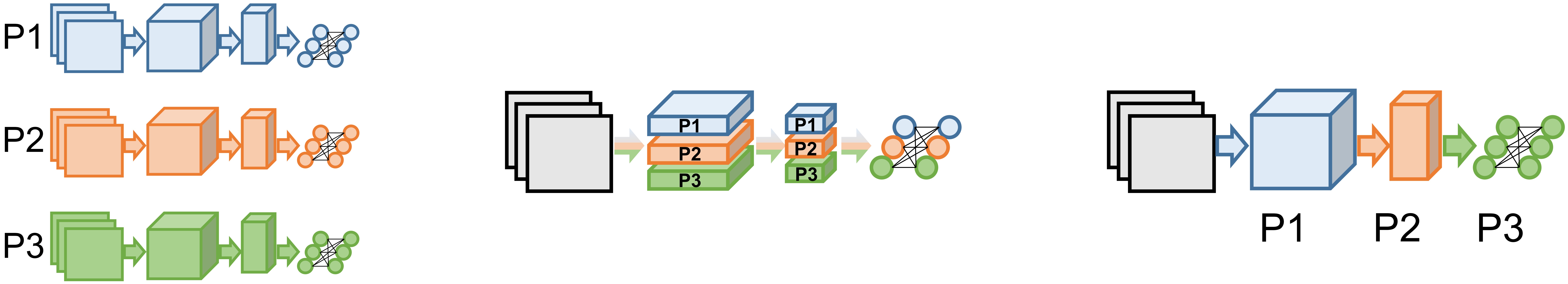


Figure 1: The transformer block structure of Megatron

Mike Isaev (GT Ph.D.), **Nic McDonald** (NVIDIA), **L. Dennison** (NVIDIA), **R. Vuduc** (unpublished 2023 manuscript)



Mike Isaev (GT Ph.D.), **Nic McDonald** (NVIDIA), **L. Dennison** (NVIDIA), **R. Vuduc** (unpublished 2023 manuscript)



(a) Data Parallelism (b) Model Parallelism (c) Layer Pipelining

Fig. 14. Neural Network Parallelism Schemes

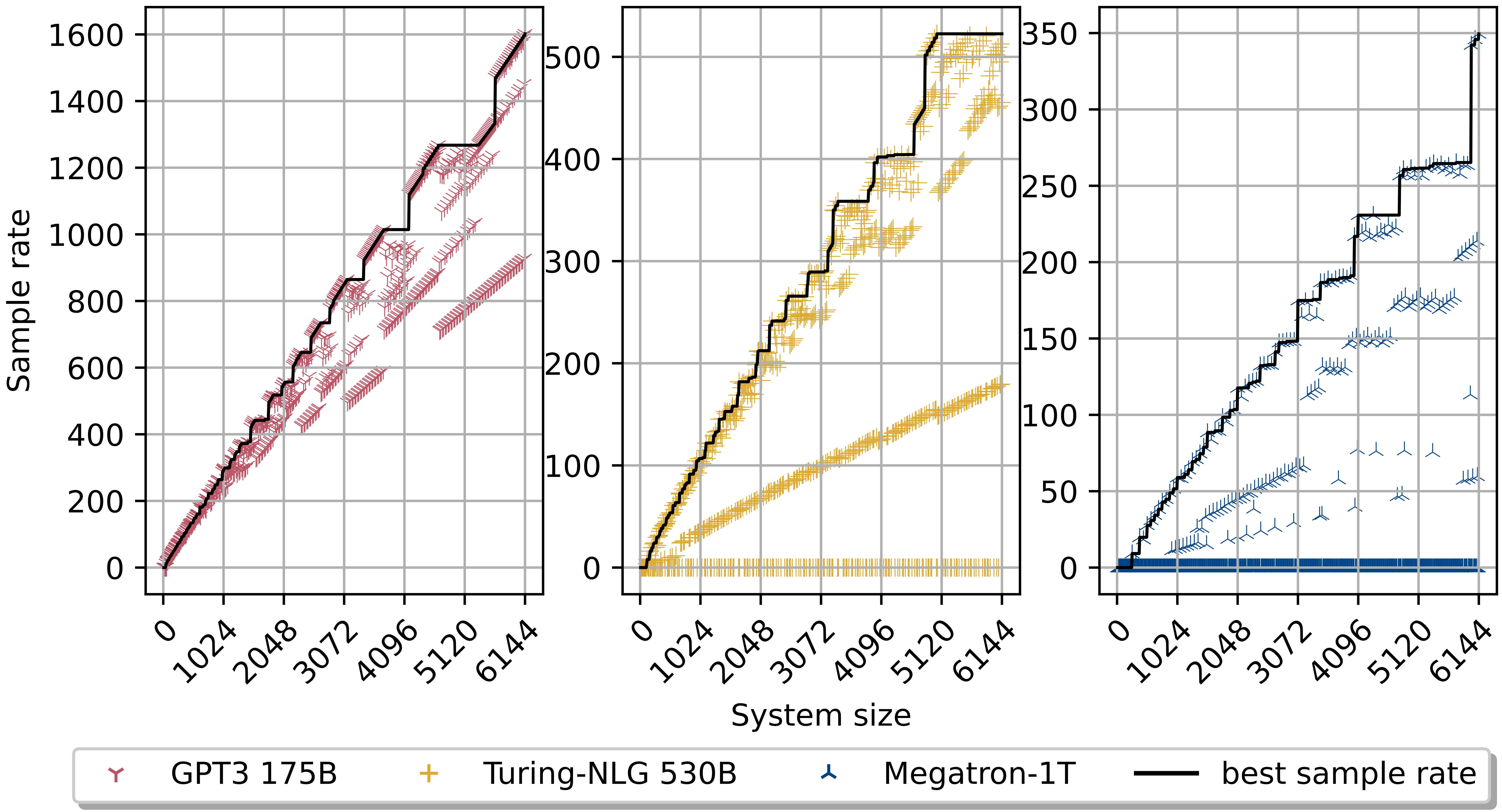
Ben-Nun & Hoefler (2019). "Demystifying parallel and distributed deep learning: an in-depth concurrency analysis."

doi:10.1145/3320060

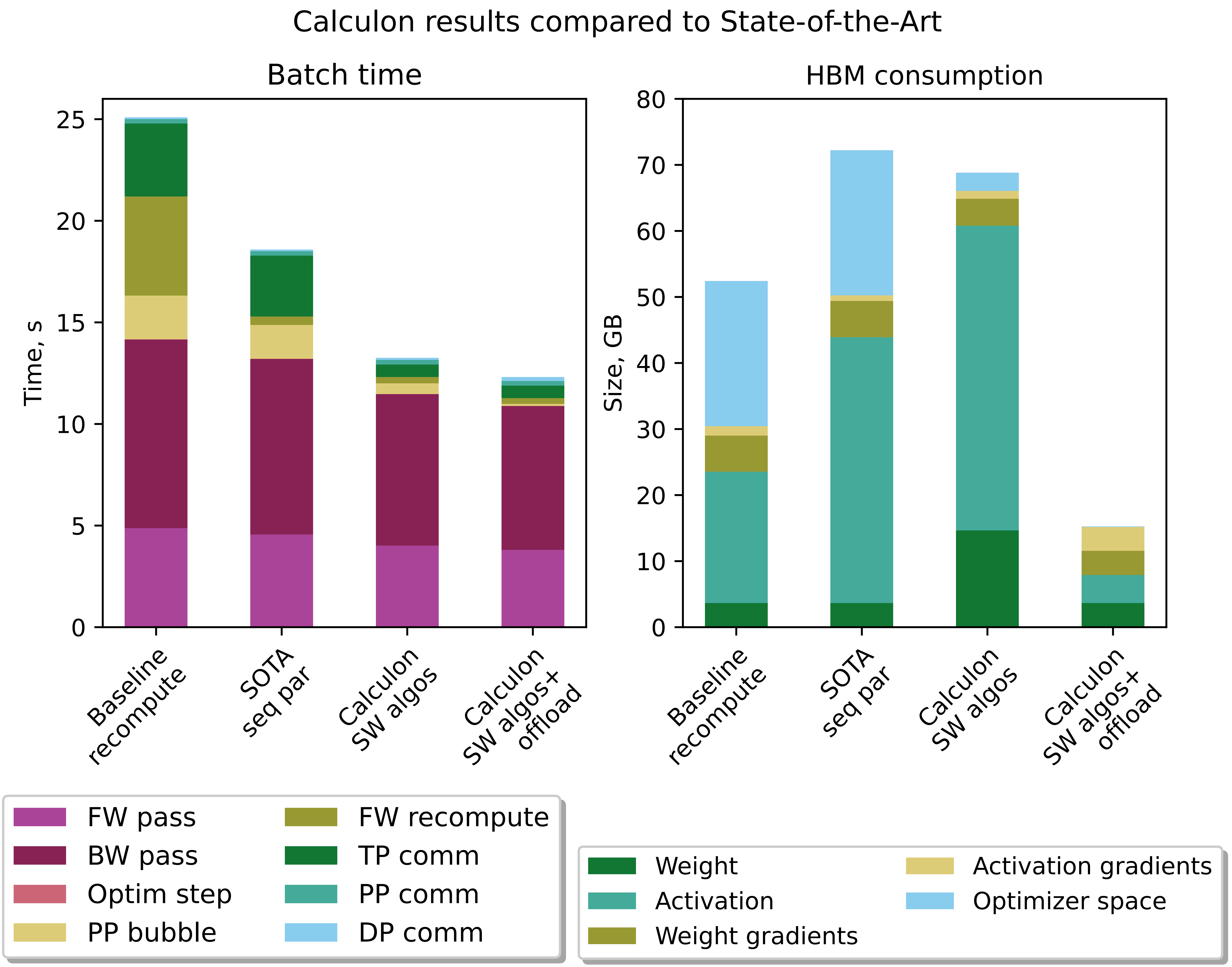
Optimization	**Year**	**Related system**	**Comp time**	**Comp util**	**Mem time**	**Mem cap**	**Mem BW**	**Net time**	**Net BW**	**range**
Data parallelism (DP) [61]	1989	network	–	↑	–	↑↑↑	–	↑	↑	1 .. batch
DP overlap [25]	2017	network	↑	↓	–	–	–	↓↓↓	–	true/false
Optimizer sharding [24]	2019	network	↓	–	–	↓↓	–	–	–	true/false
Recompute [5, 10]	2000	compute	↑↑	–	–	↓↓↓	–	–	–	full/attn/none
Fused layers [28]	2018	compute	–	↑↑	↓↓	↓↓	↓	–	–	true/false
Microbatch training [13]	2019	compute	–	↑↑	–	↑↑↑	–	–	–	1 .. batch/DP
Pipeline parallelism (PP) [7, 13]	2012	network	↑	↓↓	–	↓↓	–	↑	↑	1 .. blocks
PP 1F1B schedule [7, 32]	2012	network	–	–	–	↓↓	–	–	–	true/false
PP interleaving [33]	2021	network	↓	↑↑	–	↑	–	↑	↑↑	1 .. blocks/PP
PP RS + AG [21]	2022	network	–	–	–	–	–	↓	↓↓	true/false
Tensor parallelism (TP) [7, 22, 49]	2012	network	↓↓	↓	–	↓↓	↓↓	↑↑↑	↑↑↑	1 .. attn
TP RS + AG instead AR [33]	2021	network	–	–	↑	↑	–	↓	↓	true/false
Sequence parallelism (SP) [21]	2022	network	↓	–	↓	↓↓	↓	↑	↑	true/false
TP redo for SP [21]	2022	network	–	–	–	↓	–	↑	↑	true/false
TP overlap [58]	2022	network	↑	↓	–	–	–	↓↓	–	true/false
Weight offload [48]	2021	memory	–	–	↑	↓↓↓	↑	–	–	true/false
Activation offload [48]	2021	memory	–	–	↑	↓↓↓	↑	–	–	true/false
Optimizer offload [48]	2021	memory	–	–	↑	↓	↑	–	–	true/false

Mike Isaev (GT Ph.D.), **Nic McDonald** (NVIDIA), **L. Dennison** (NVIDIA), **R. Vuduc** (unpublished 2023 manuscript)

LLM training scalability

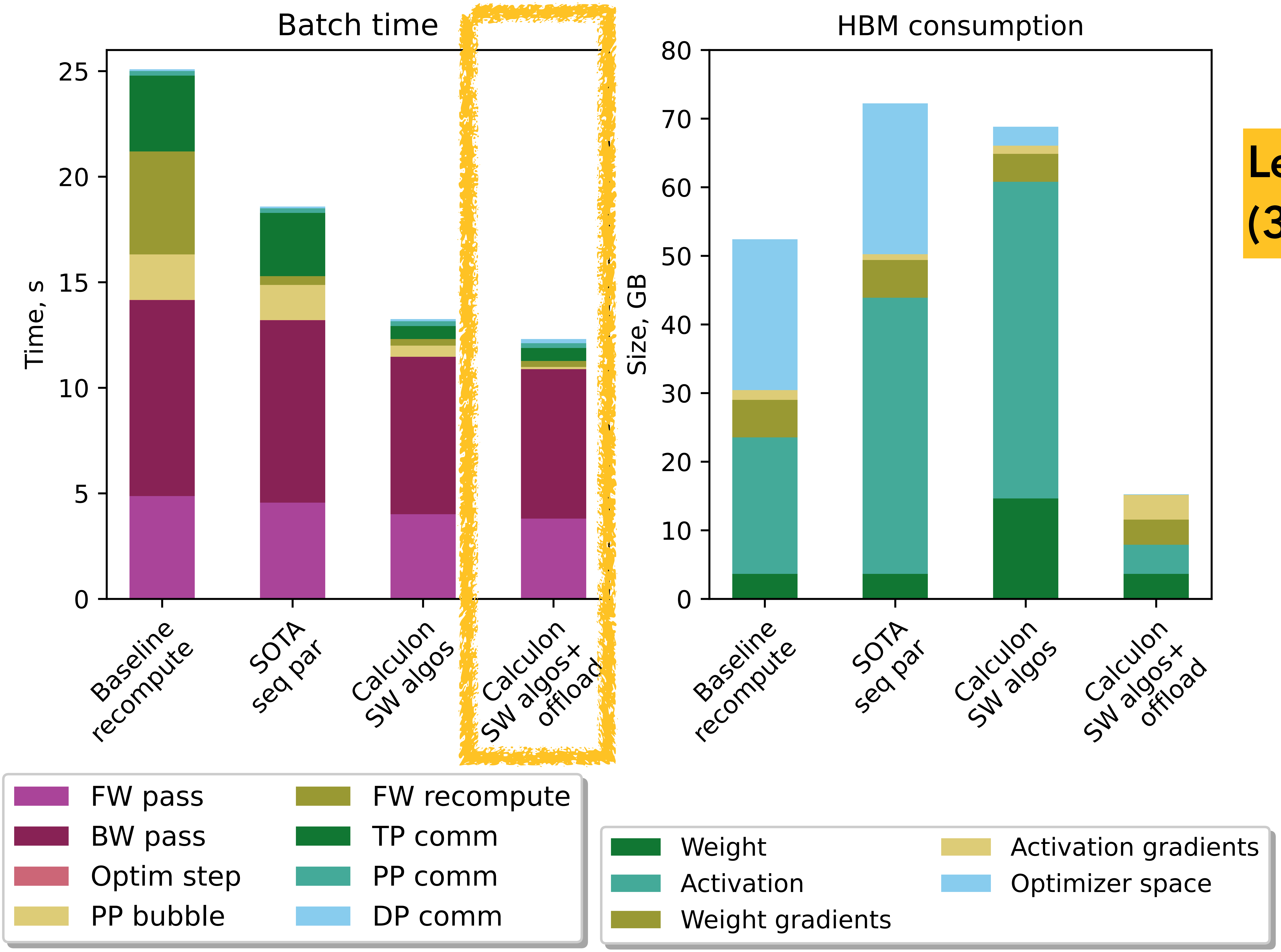


Mike Isaev (GT Ph.D.), **Nic McDonald** (NVIDIA), **L. Dennison** (NVIDIA), **R. Vuduc** (unpublished 2023 manuscript)



Mike Isaev (GT Ph.D.), **Nic McDonald** (NVIDIA), **L. Dennison** (NVIDIA), **R. Vuduc** (unpublished 2023 manuscript)

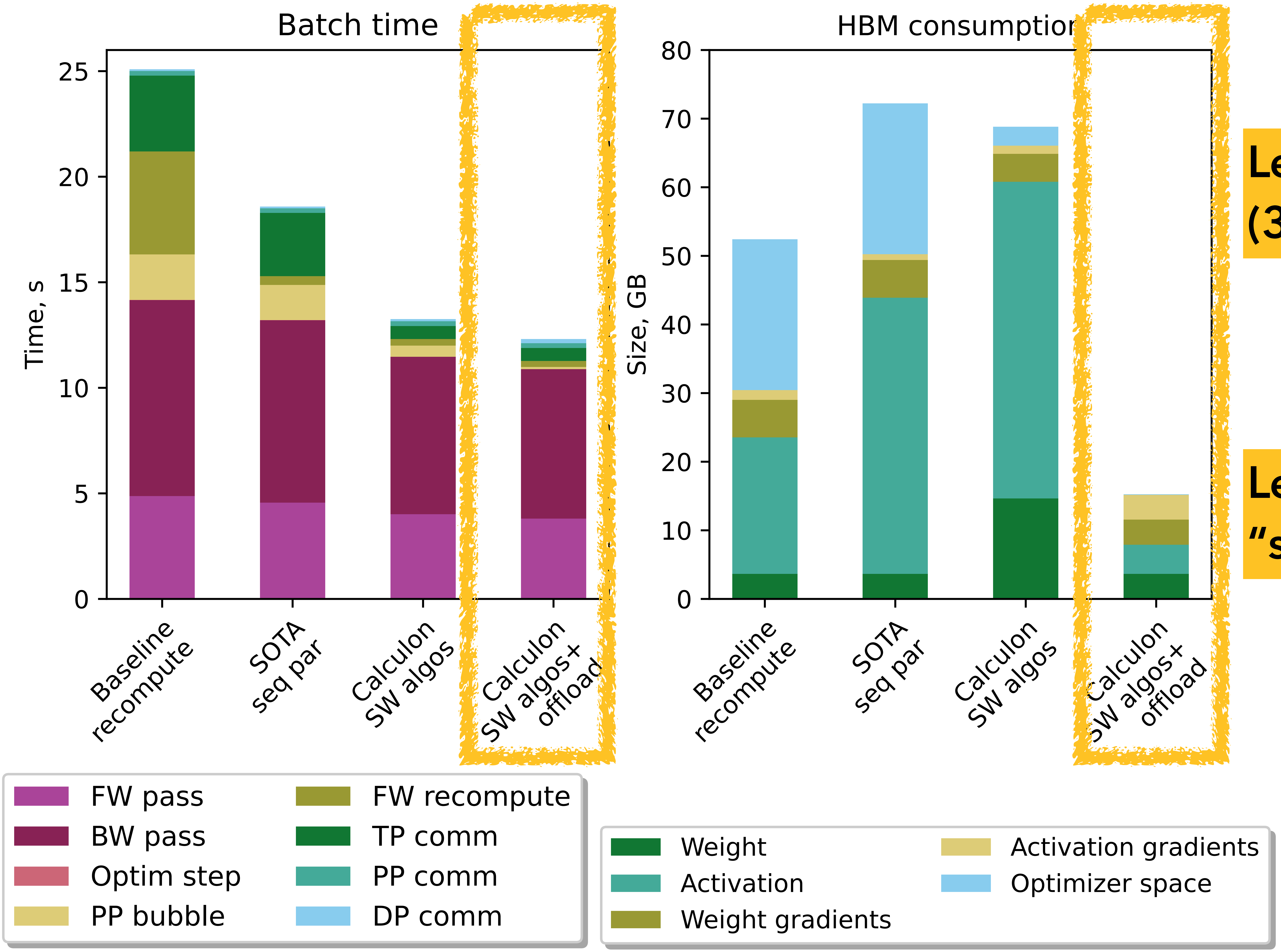
Calculon results compared to State-of-the-Art



Less time, higher MFU
(38% → 75%)

Mike Isaev (GT Ph.D.), **Nic McDonald** (NVIDIA), **L. Dennison** (NVIDIA), **R. Vuduc** (unpublished 2023 manuscript)

Calculon results compared to State-of-the-Art



Less time, higher MFU (38% → 75%)

Less HBM – trade for more "slow" memory

Mike Isaev (GT Ph.D.), **Nic McDonald** (NVIDIA), **L. Dennison** (NVIDIA), **R. Vuduc** (unpublished 2023 manuscript)

Q: Is the cloud for everyone, e.g., HPC people?

A: Sure, why not?

Q: What system will the cloud provide?

Q: What is an optimal GPU machine for DL?

A: Add slow memory, "modest" networks

Great! So what's the problem?

Recall:

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

Reduces **energy**: fewer flops, less storage

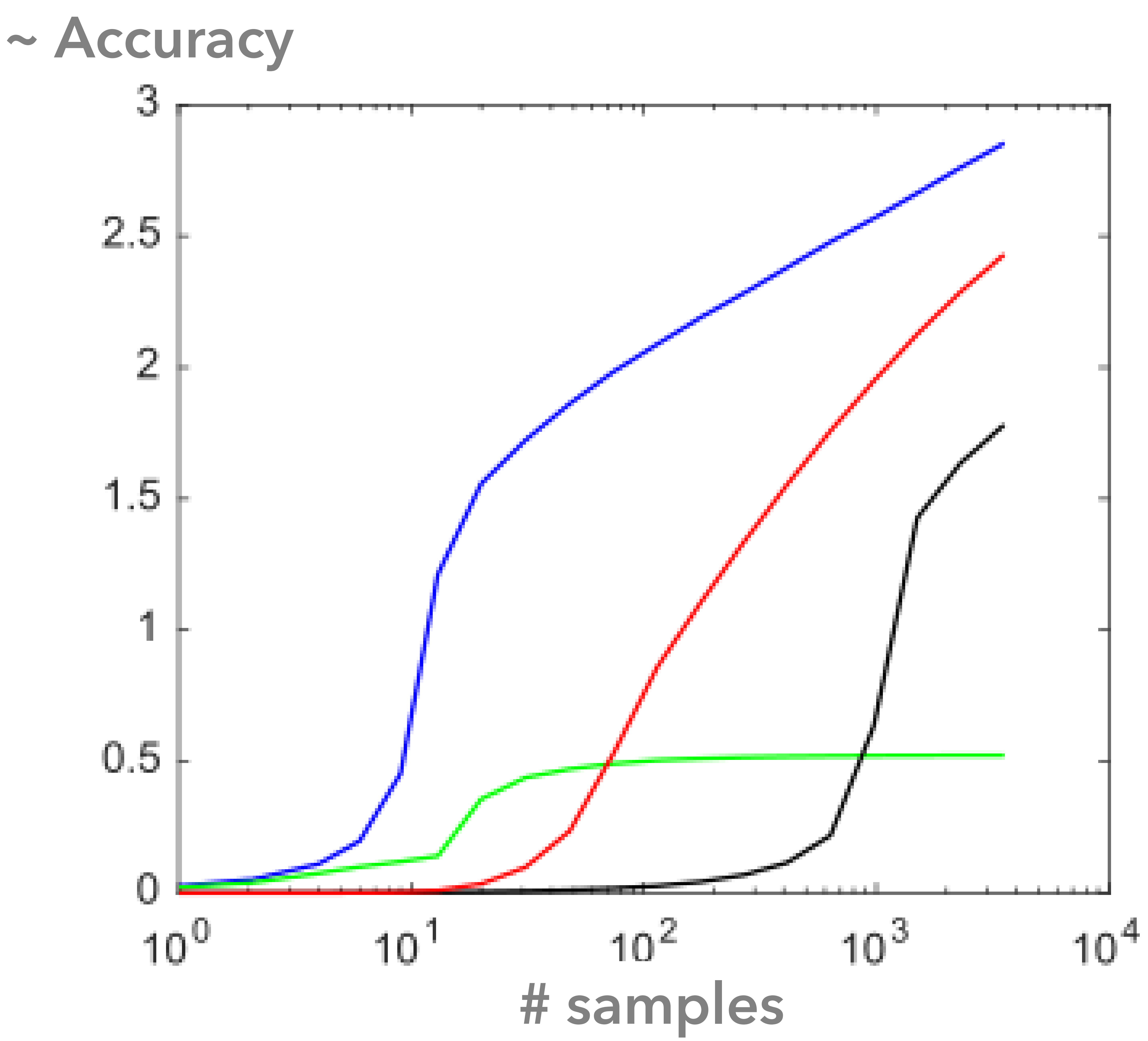
Recall:

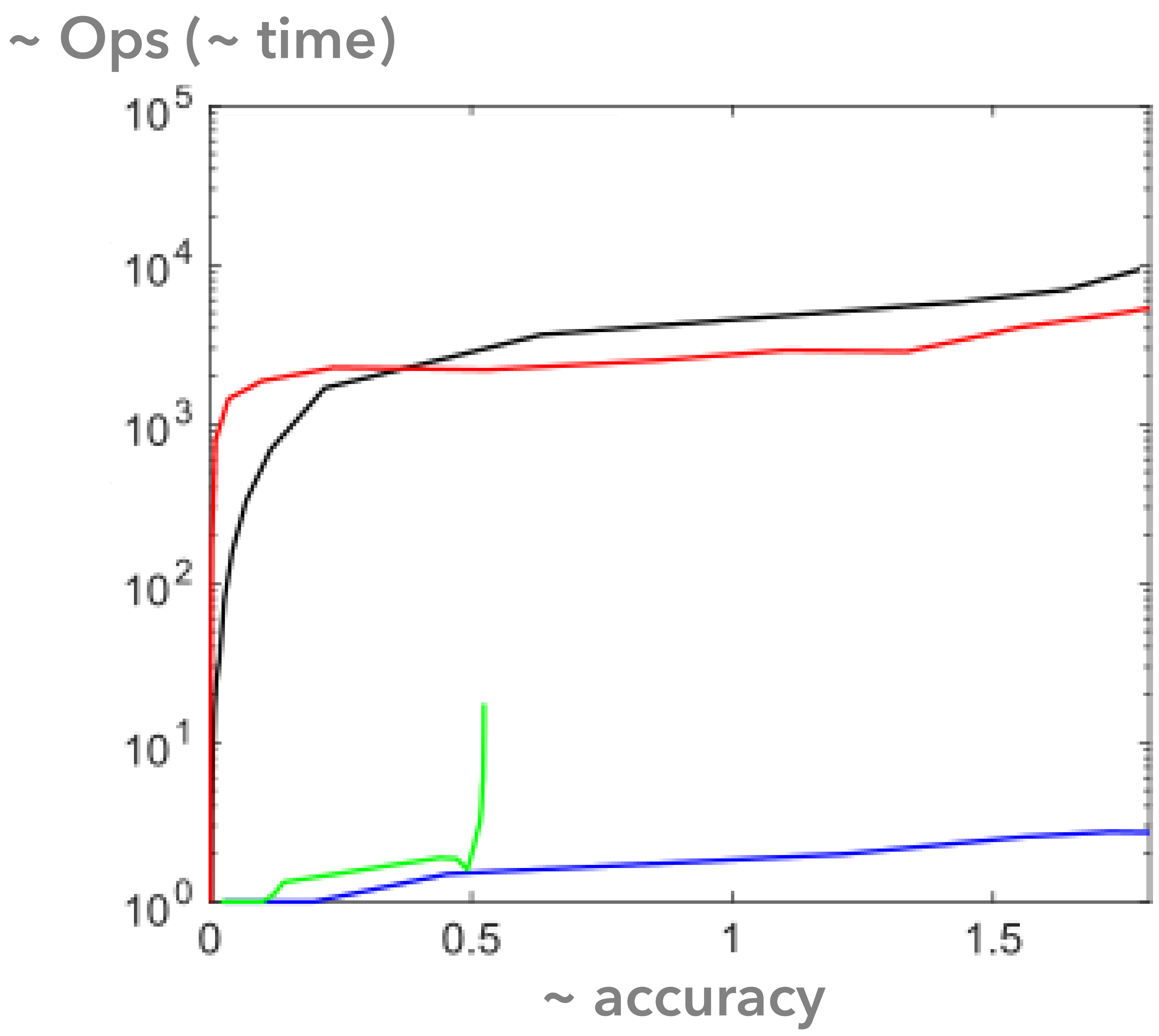
$$\mathcal{O}(N^2) \longrightarrow \boxed{\mathcal{O}(N)}$$

% time communicating increases

Multiple regression

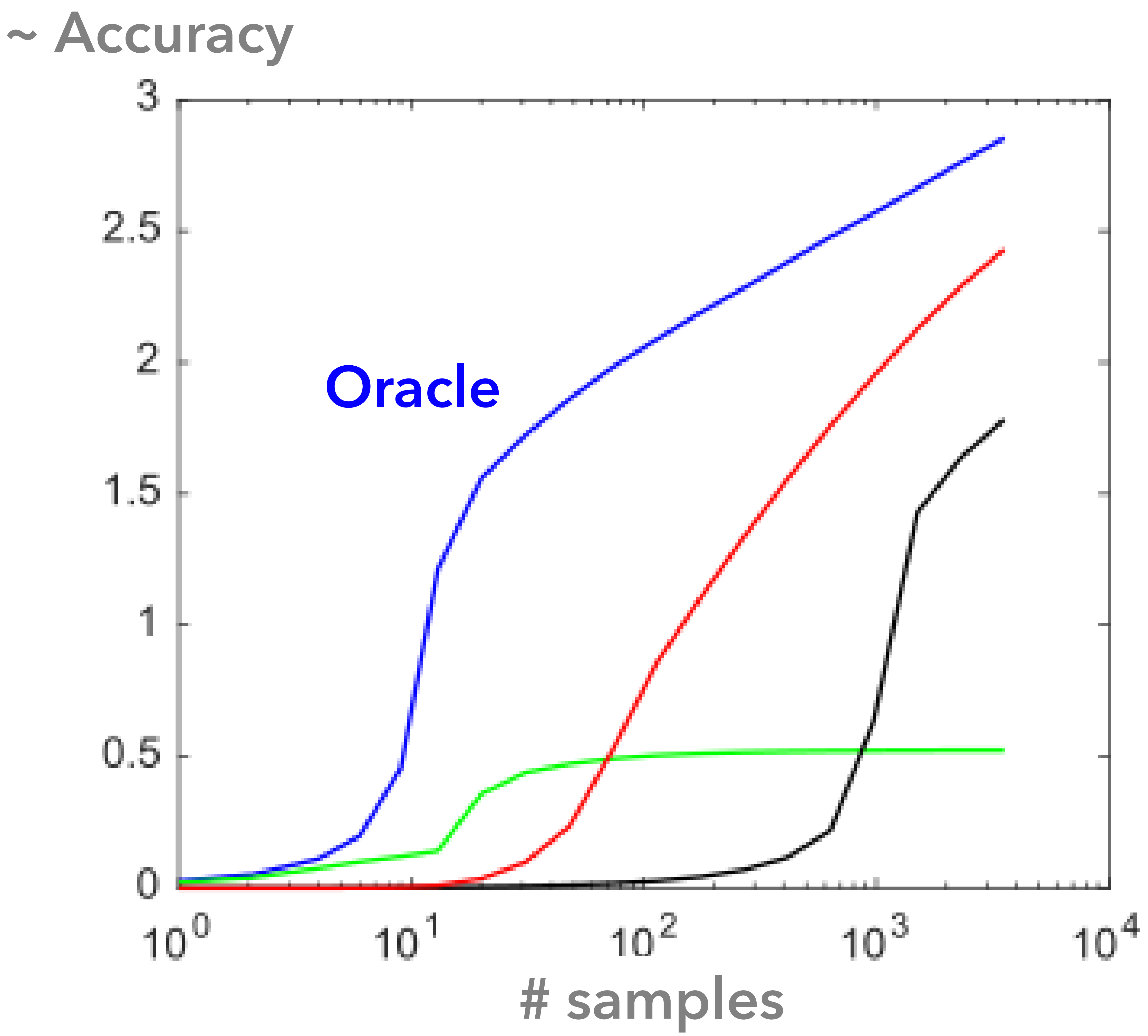
$$X \equiv \begin{bmatrix} \big| & \big| & \big| & & \big| \\ x_{i,0} & x_{i,1} & x_{i,2} & \cdots & x_{i,n-1} \\ \Big\downarrow & \Big\downarrow & \Big\downarrow & & \Big\downarrow \end{bmatrix}$$

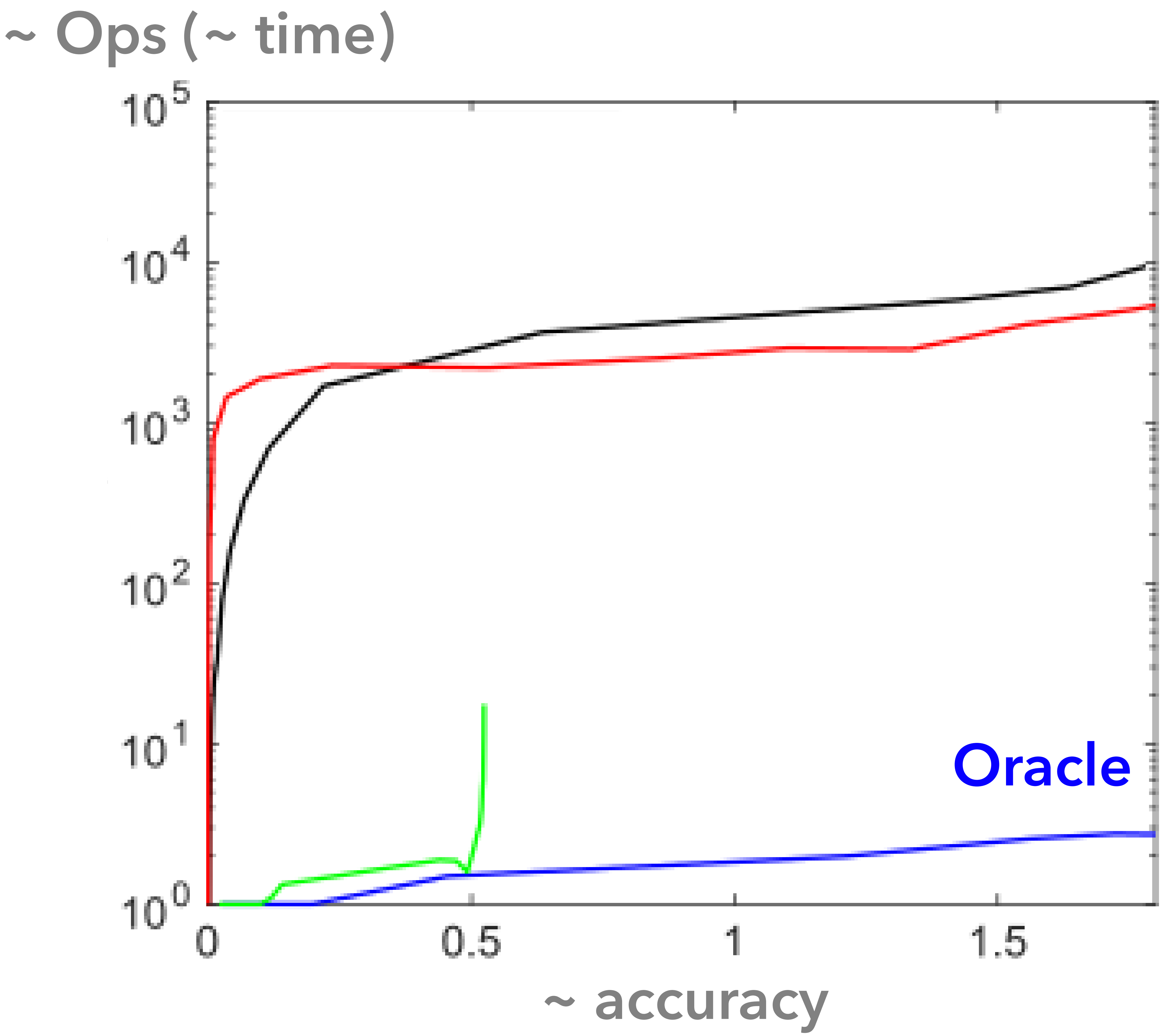




Linear regression example: Thompson et al., "The computational limits of deep learning" (July 2020). arXiv:2007.05558v1

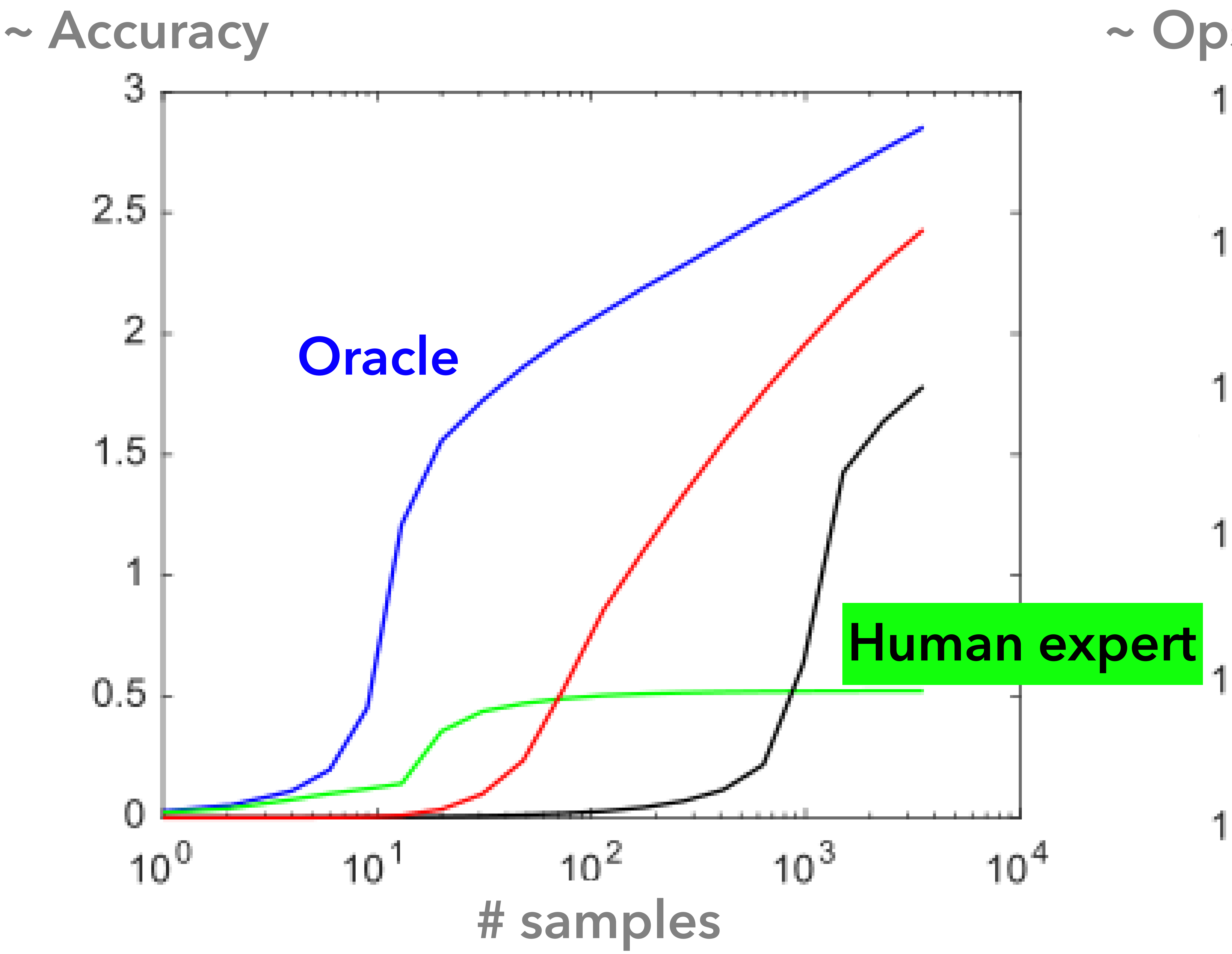
More samples → More accuracy, reasonable time

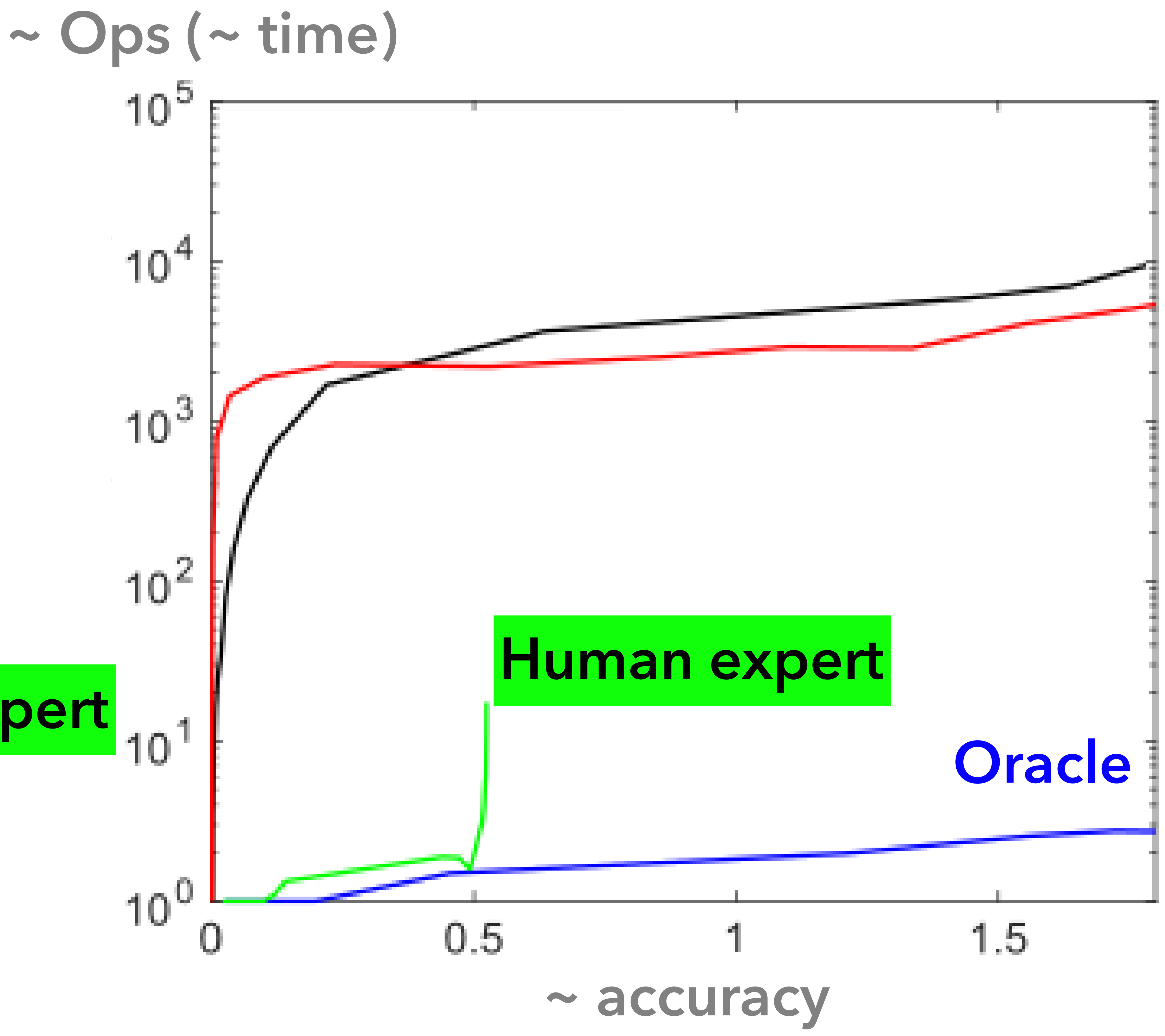




Linear regression example: Thompson et al., “The computational limits of deep learning” (July 2020). arXiv:2007.05558v1

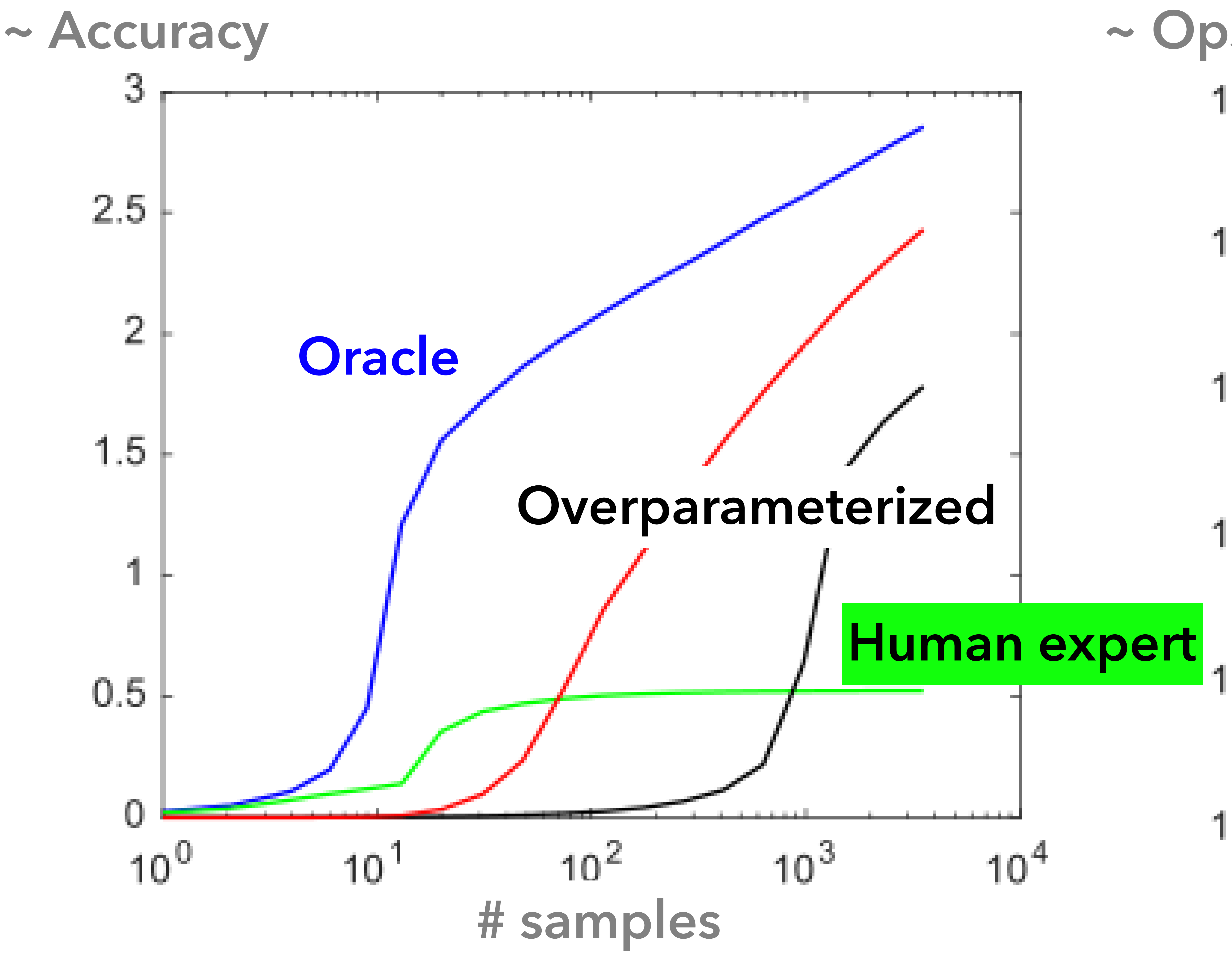
Accuracy plateaus and costs rise

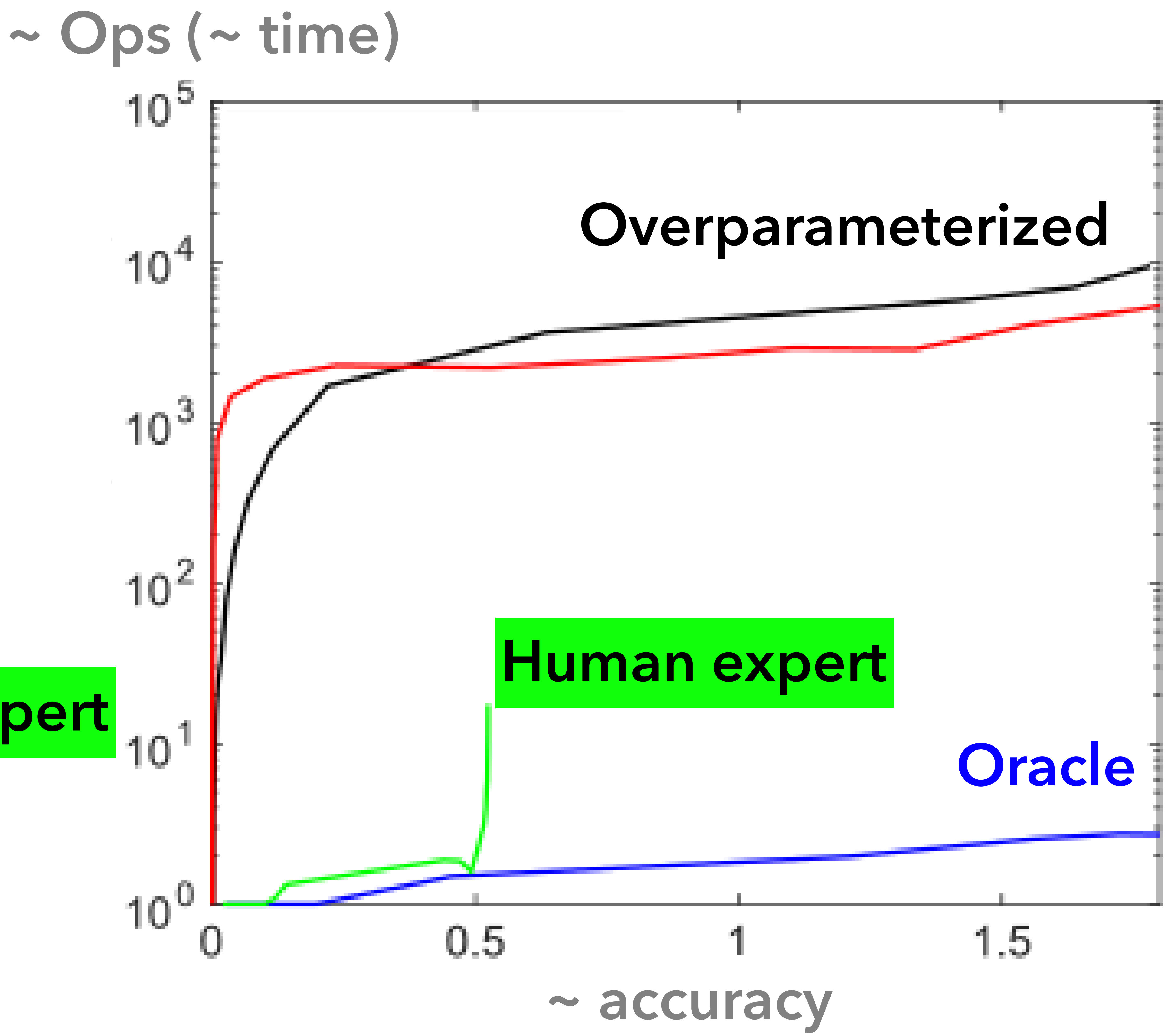




Linear regression example: Thompson et al., "The computational limits of deep learning" (July 2020). arXiv:2007.05558v1

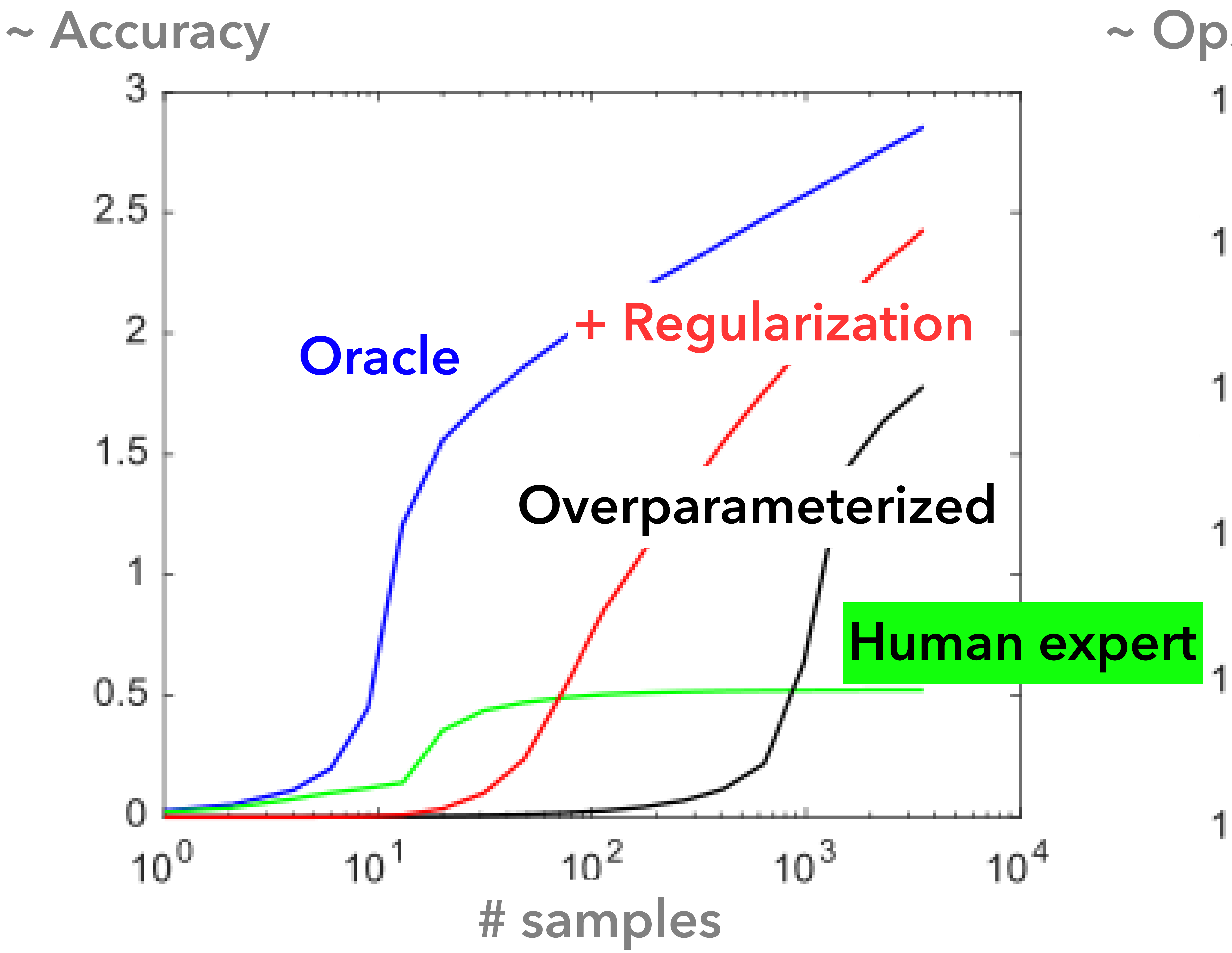
With enough data, more accuracy but a high cost

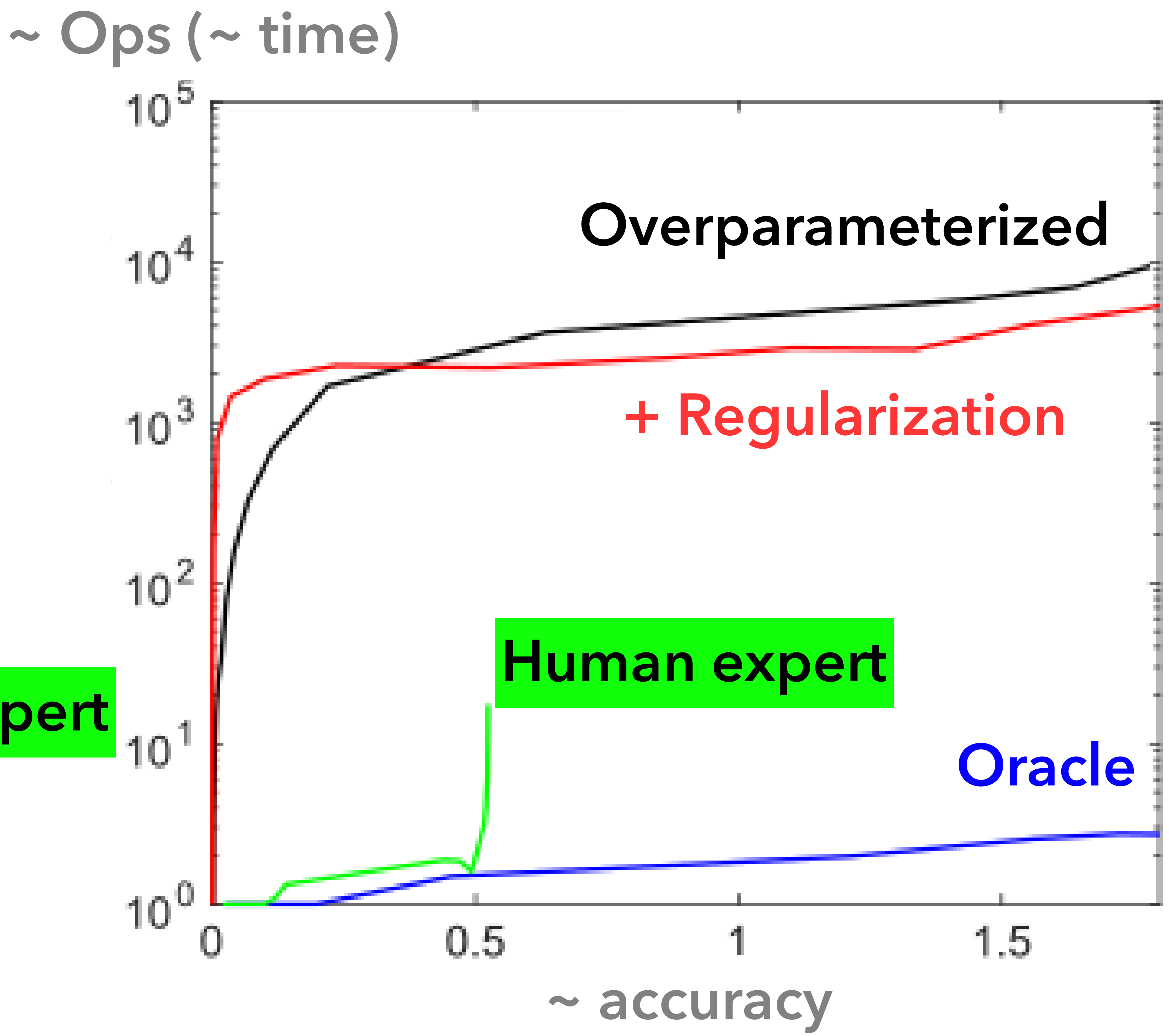




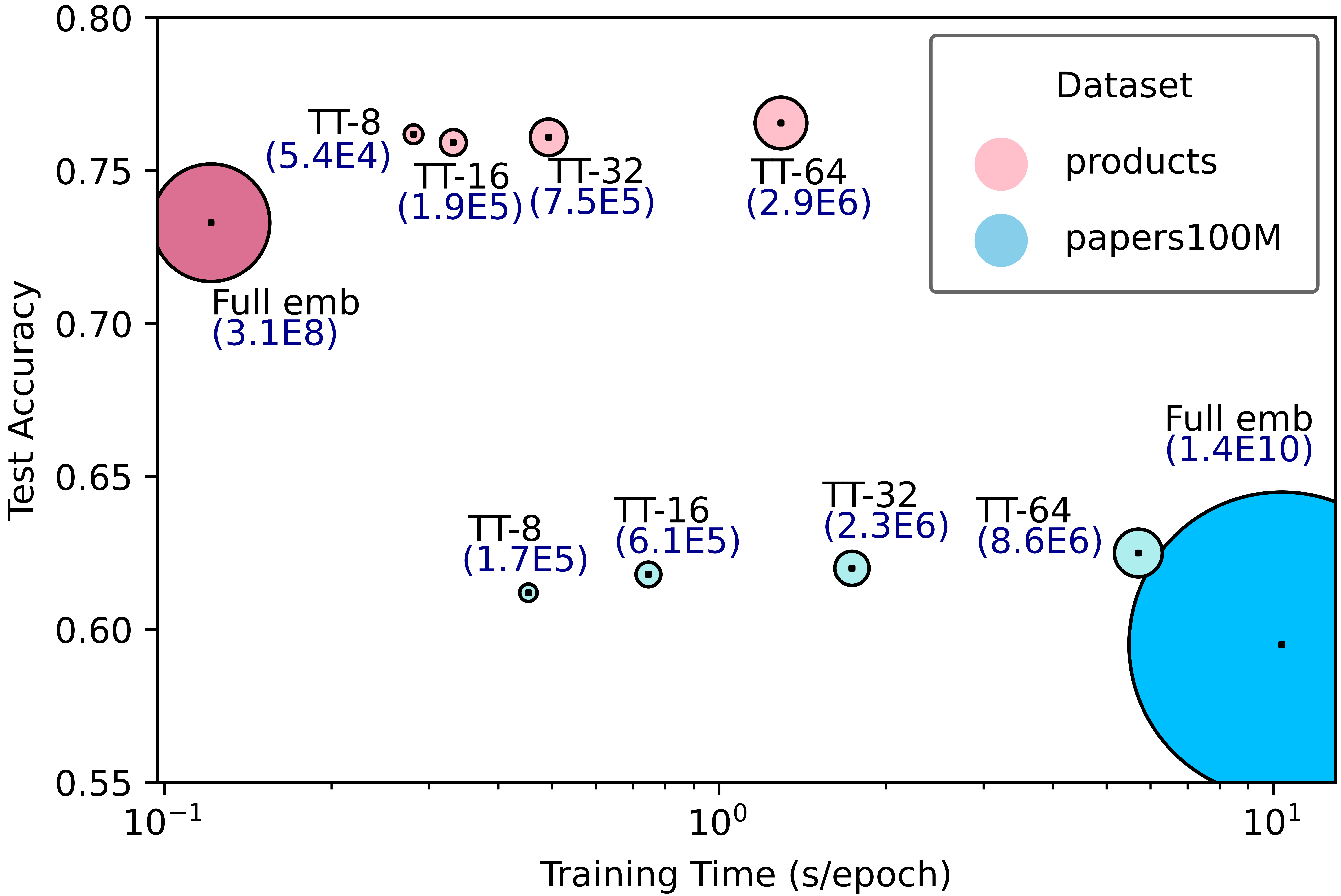
Linear regression example: Thompson et al., “The computational limits of deep learning” (July 2020). arXiv:2007.05558v1

Better accuracy with fewer samples, but still expensive



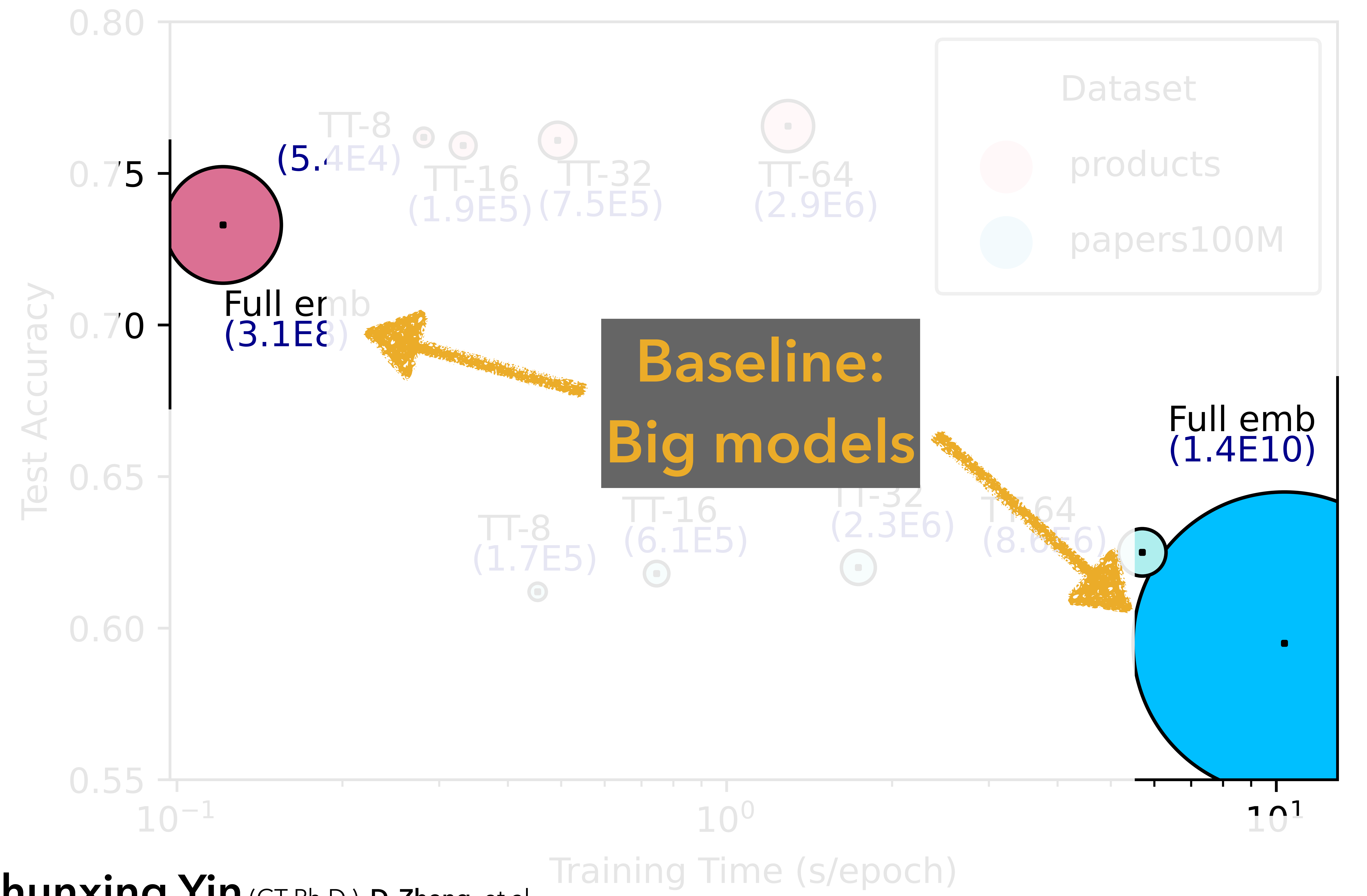


Linear regression example: Thompson et al., "The computational limits of deep learning" (July 2020). arXiv:2007.05558v1



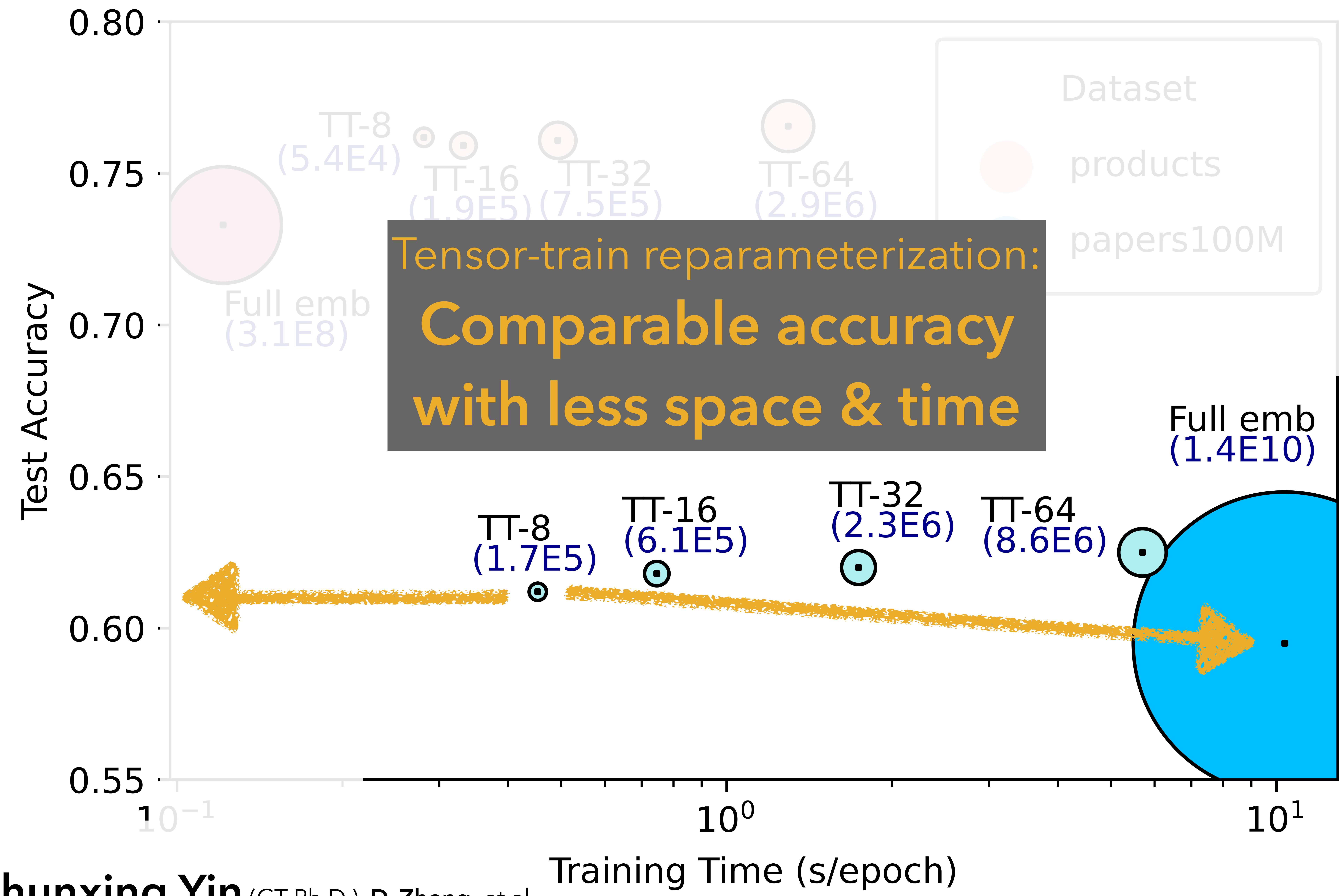
Chunxing Yin (GT Ph.D.), **D. Zheng** (Amazon), I. Nisrat, C. Faloutsos, G. Karypis, R. Vuduc.

"Nimble GNN embedding with **tensor-train decomposition**." In *KDD'22*. doi:10.1145/3534678.3539423



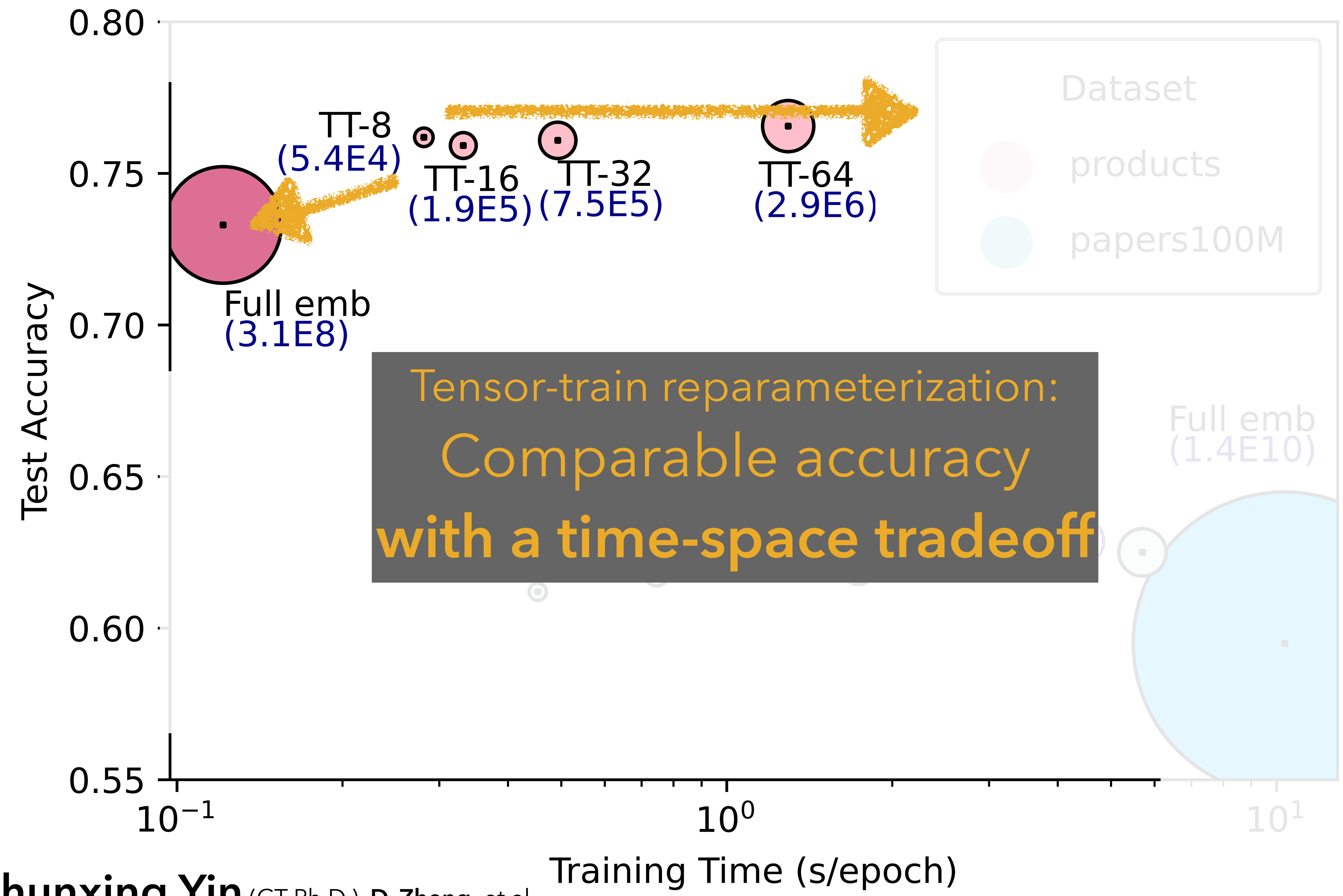
Chunxing Yin (GT Ph.D.), **D. Zheng,** et al.

"Nimble GNN embedding with **tensor-train decomposition**." In *KDD'22*. doi:10.1145/3534678.3539423



Chunxing Yin (GT Ph.D.), **D. Zheng,** et al.

"Nimble GNN embedding with **tensor-train decomposition**." In *KDD'22*. doi:10.1145/3534678.3539423



Tensor-train reparameterization:

Comparable accuracy

with a time-space tradeoff

Chunxing Yin (GT Ph.D.), **D. Zheng,** et al.

"Nimble GNN embedding with **tensor-train decomposition**." In *KDD'22*. doi:10.1145/3534678.3539423

Algorithms for 2D Poisson Equation with N unknowns

Algorithm	Serial	PRAM	Memory	#Procs	(Keyes '04)
° Dense LU	N^3	N	N^2	N^2	
° Band LU	N^2	N	$N^{3/2}$	N	1947
° Jacobi	N^2	N	N	N	1950
° Explicit Inv.	N^2	log N	N^2	N^2	
° Conj.Grad.	$N^{3/2}$	$N^{1/2}$ *log N	N	N	1971
° RB SOR	$N^{3/2}$	$N^{1/2}$	N	N	
° Sparse LU	$N^{3/2}$	$N^{1/2}$	N*log N	N	~ 1970s
° FFT	N*log N	log N	N	N	
° Multigrid	N	$\log^2 N$	N	N	1984
° Lower bound	N	log N	N		

PRAM is an idealized parallel model with zero cost communication

https://sites.google.com/lbl.gov/cs267-spr2023

Q: Is the cloud for everyone, e.g., HPC people?
A: Sure, why not?

Q: What system will the cloud provide?
Q: What is an optimal GPU machine for DL?
A: Add slow memory, “modest” networks

Q: Are we overfitting?

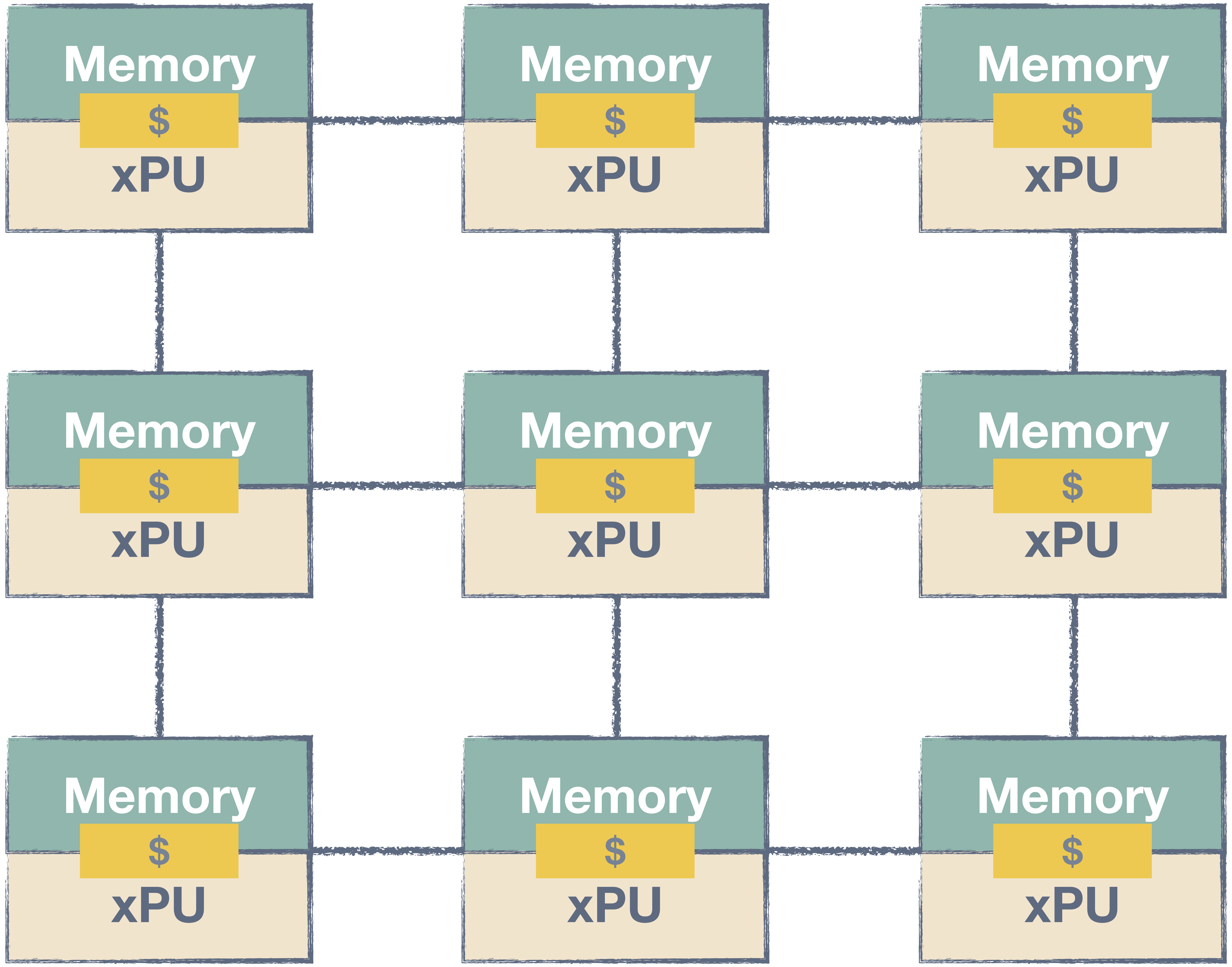
An Iron Law of Parallel and Distributed Computation

A modern cluster or supercomputer is, to first order, a collection of processing nodes. Each node has a processor ("xPU") and a two-level memory hierarchy. Nodes are connected by a network.

As a program executes on this system, it incurs two types of communication cost.

"Vertical" communication occurs in the memory system between, say, RAM and cache.

"Horizontal" communication occurs between nodes across the network.



Two costs: $T_{\text{network}} + T_{\text{memory}}$

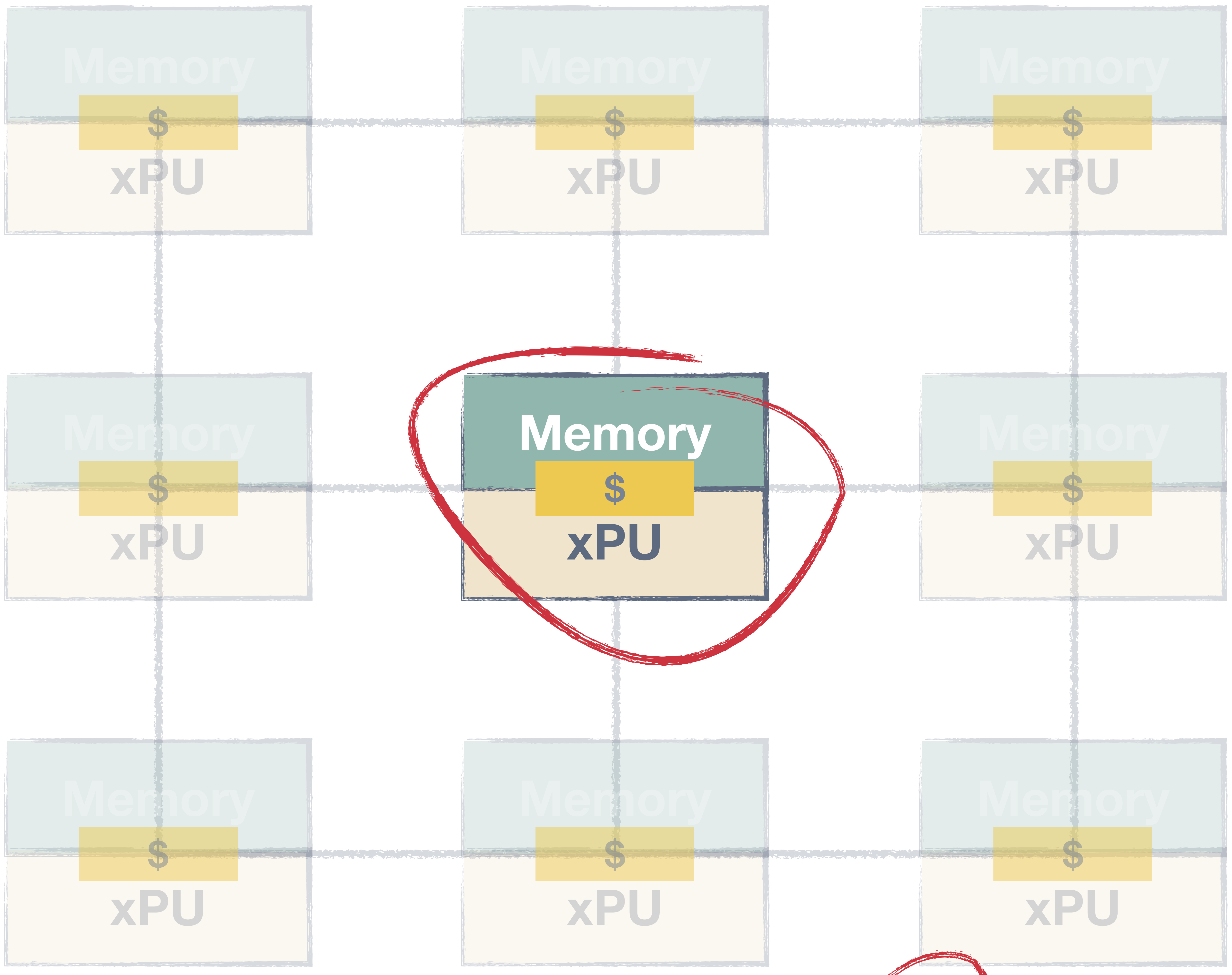
An Iron Law of Parallel and Distributed Computation

A modern cluster or supercomputer is, to first order, a collection of processing nodes. Each node has a processor ("xPU") and a two-level memory hierarchy. Nodes are connected by a network.

As a program executes on this system, it incurs two types of communication cost.

"Vertical" communication occurs in the memory system between, say, RAM and cache.

"Horizontal" communication occurs between nodes across the network.



Two costs: $T_{\text{network}} + T_{\text{memory}}$

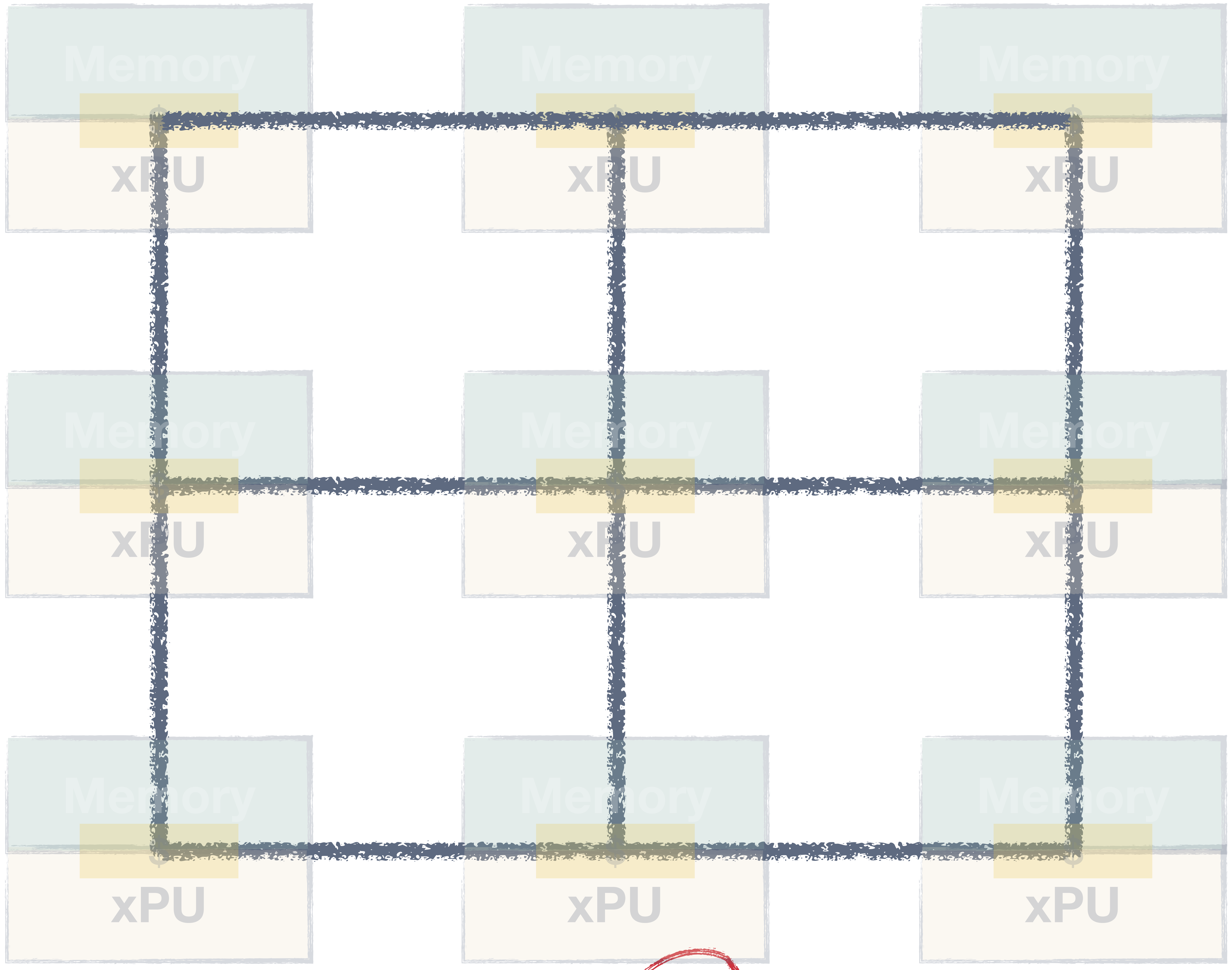
An Iron Law of Parallel and Distributed Computation

A modern cluster or supercomputer is, to first order, a collection of processing nodes. Each node has a processor ("xPU") and a two-level memory hierarchy. Nodes are connected by a network.

As a program executes on this system, it incurs two types of communication cost.

"Vertical" communication occurs in the memory system between, say, RAM and cache.

"Horizontal" communication occurs between nodes across the network.



Two costs: $T_{network} + T_{memory}$

(Asymptotic running time – rules-of-thumb)

Jeff Young (GT), R.V. (2016) – "Finding balance in post-Moore's Law era." [link]

(Asymptotic running time – rules-of-thumb)

Compute time

$$\frac{W(n)}{P}$$

Jeff Young (GT), R.V. (2016) – "Finding balance in post-Moore's Law era." [link]

(Asymptotic running time – rules-of-thumb)

Compute time

$$\frac{W(n)}{P}$$

P-fold speedup, ideally

Jeff Young (GT), R.V. (2016) – "Finding balance in post-Moore's Law era." [link]

(Asymptotic running time – rules-of-thumb)

Compute time	**Memory time**
$\frac{W(n)}{P}$	$\frac{W(n)}{P \cdot f(Z)}$
P-fold speedup, ideally	

Jeff Young (GT), R.V. (2016) – "Finding balance in post-Moore's Law era." [link]

(Asymptotic running time – rules-of-thumb)

Compute time	**Memory time**
$$\frac{W(n)}{P}$$	$$\frac{W(n)}{P \cdot f(Z)}$$
P-fold speedup, ideally	**e.g.,** $\sqrt{Z}$, $\log Z$

Jeff Young (GT), R.V. (2016) – "Finding balance in post-Moore's Law era." [link]

(Asymptotic running time – rules-of-thumb)

Compute time	Memory time	Network time
$$\frac{W(n)}{P}$$	$$\frac{W(n)}{P \cdot f(Z)}$$	$$\frac{W(n)}{h(n)} \cdot \frac{g(P)}{P}$$
P-fold speedup, ideally	e.g., $\sqrt{Z}$, $\log Z$	

Jeff Young (GT), R.V. (2016) – "Finding balance in post-Moore's Law era." [link]

(Asymptotic running time – rules-of-thumb)

Compute time

$$\frac{W(n)}{P}$$

P-fold speedup, ideally

Memory time

$$\frac{W(n)}{P \cdot f(Z)}$$

e.g., $\sqrt{Z}$, $\log Z$

Network time

$$\frac{W(n)}{h(n)} \cdot \frac{g(P)}{P}$$

Asymptotic reduction

Jeff Young (GT), R.V. (2016) – "Finding balance in post-Moore's Law era." [link]

(Asymptotic running time – rules-of-thumb)

Compute time

$$\frac{W(n)}{P}$$

P-fold speedup, ideally

Memory time

$$\frac{W(n)}{\boxed{P \cdot} f(Z)}$$

e.g., $\sqrt{Z}$, $\log Z$

Network time

$$\frac{W(n)}{h(n)} \cdot \frac{\boxed{g(P)}}{P}$$

Asymptotic reduction ($h(n)$)

Tradeoff ($g(P)$)

Jeff Young (GT), R.V. (2016) – "Finding balance in post-Moore's Law era." [link]

Sara Karamati

DPUs in modern clusters

The basic building block of a distributed-memory cluster or supercomputer is a node.

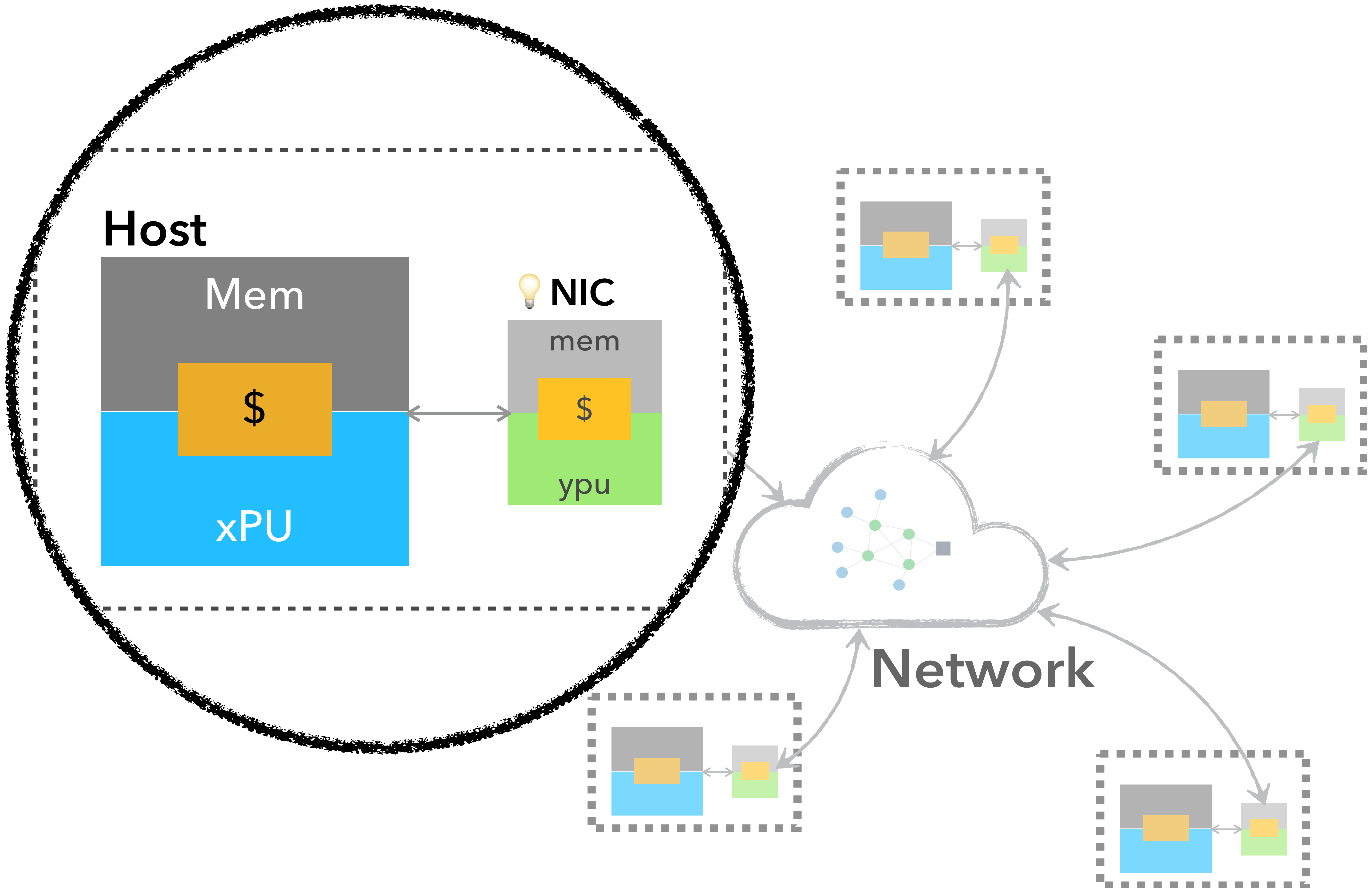
Each node includes a host, which is a processor (xPU) + memory hierarchy.

The host can communicate with other hosts via its NIC (network interface controller).

A network connects the nodes. The nodes may be arranged in some topology, which determines the network's carrying capacity and cost.

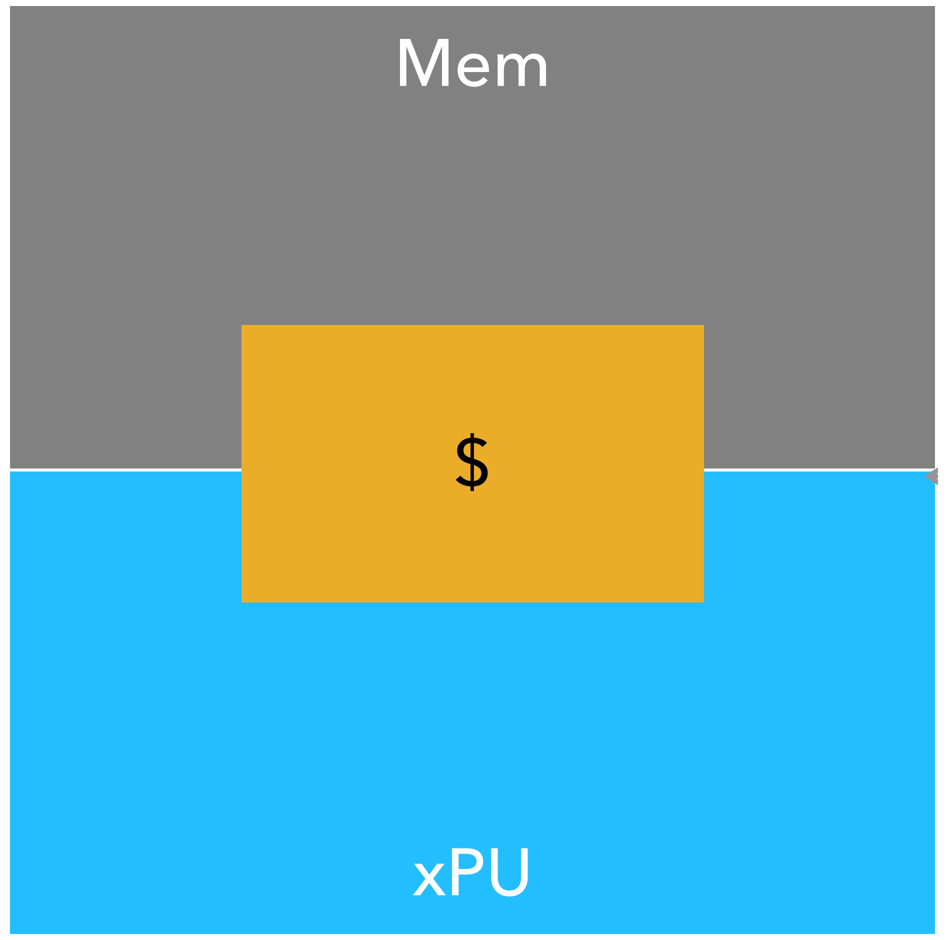
In a **smartNIC**, the NIC becomes "**host-like**" via the addition of processing (ypu) and memory.

Node



Hypothetical: Multi-SmartNIC

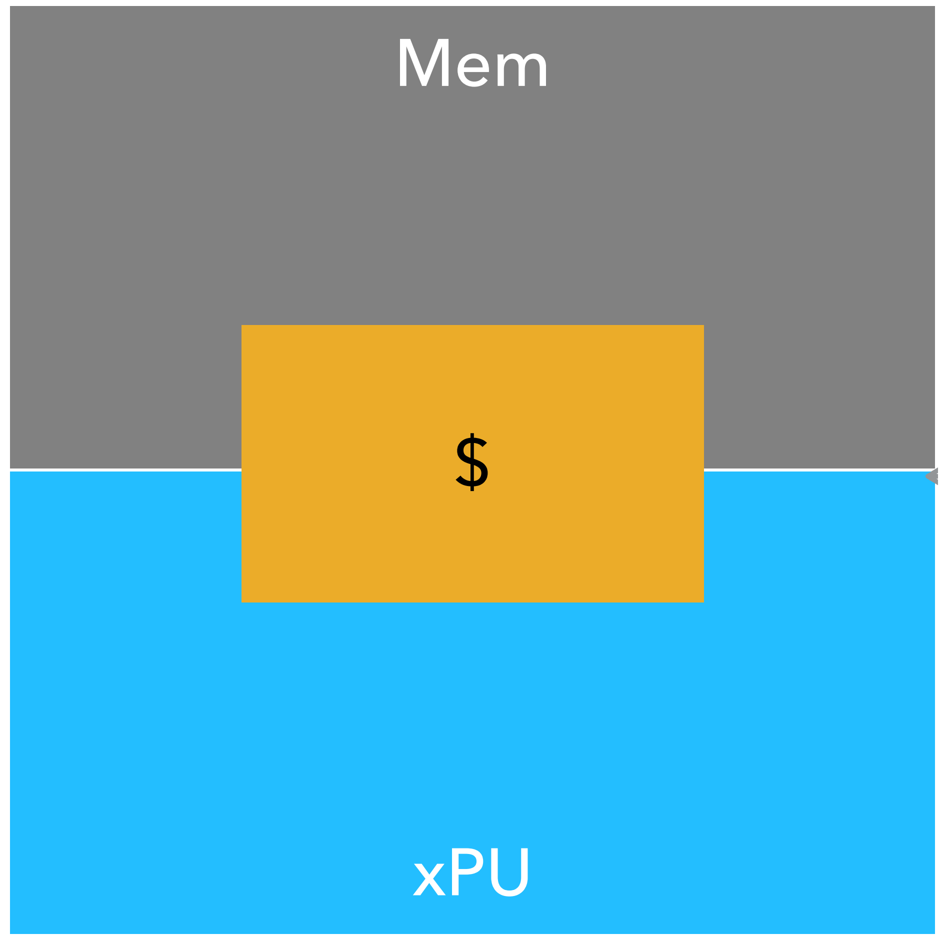
One host xPU (16 cores)



S. Karamati (GT), J. Young, R.V., et al. (2016) – “Smarter NICs for faster molecular dynamics: a case study.” doi:10.1109/IPDPS53621.2022.00063

Hypothetical: Multi-SmartNIC

One host xPU (16 cores)

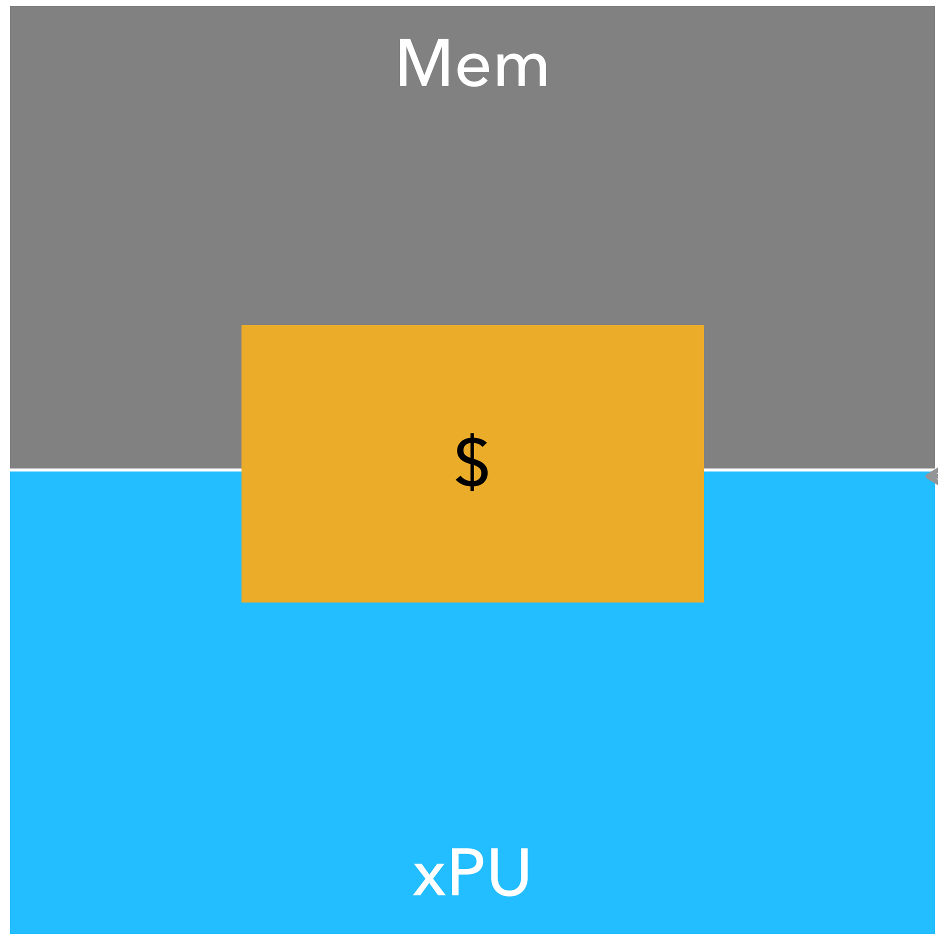


657 GF/s

S. Karamati (GT), J. Young, R.V., et al. (2016) – “Smarter NICs for faster molecular dynamics: a case study.” doi:10.1109/IPDPS53621.2022.00063

Hypothetical: Multi-SmartNIC

One host xPU (16 cores)



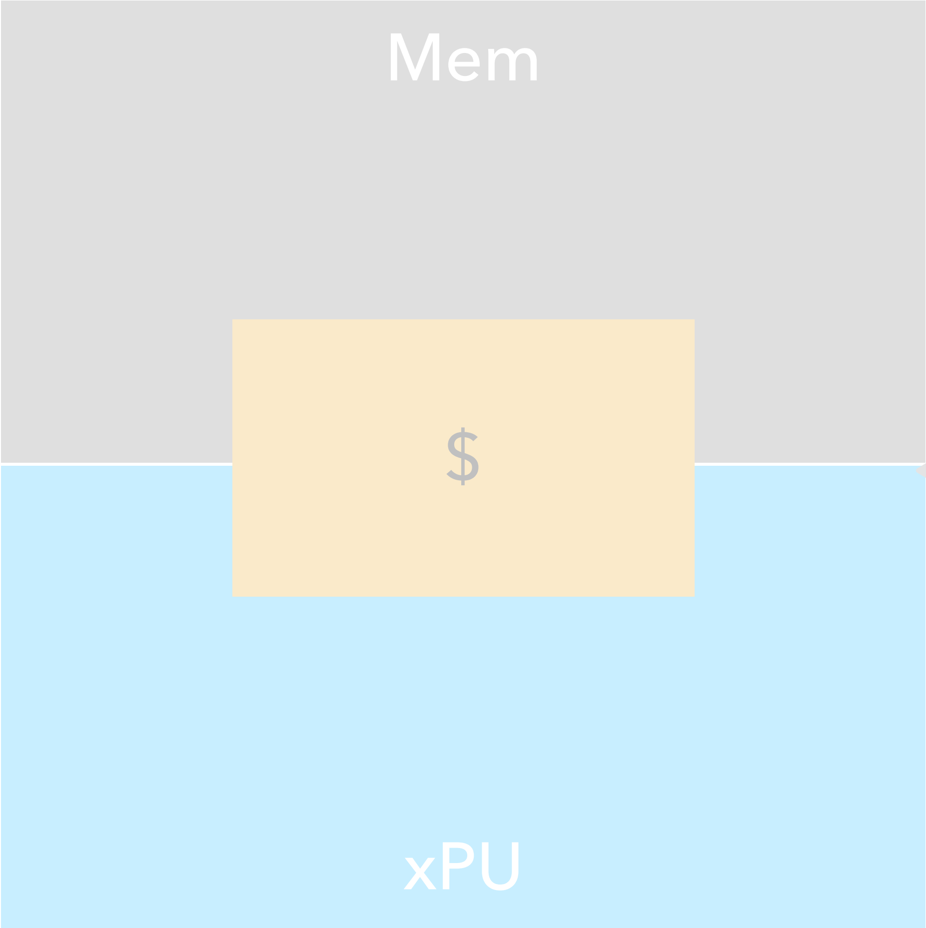
657 GF/s

76.8 GB/s

S. Karamati (GT), J. Young, R.V., et al. (2016) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

Hypothetical: Multi-SmartNIC

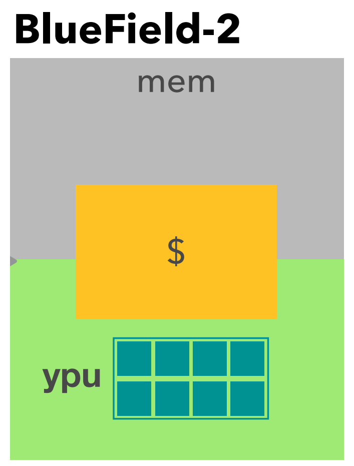
One host xPU (16 cores)



657 GF/s

76.8 GB/s

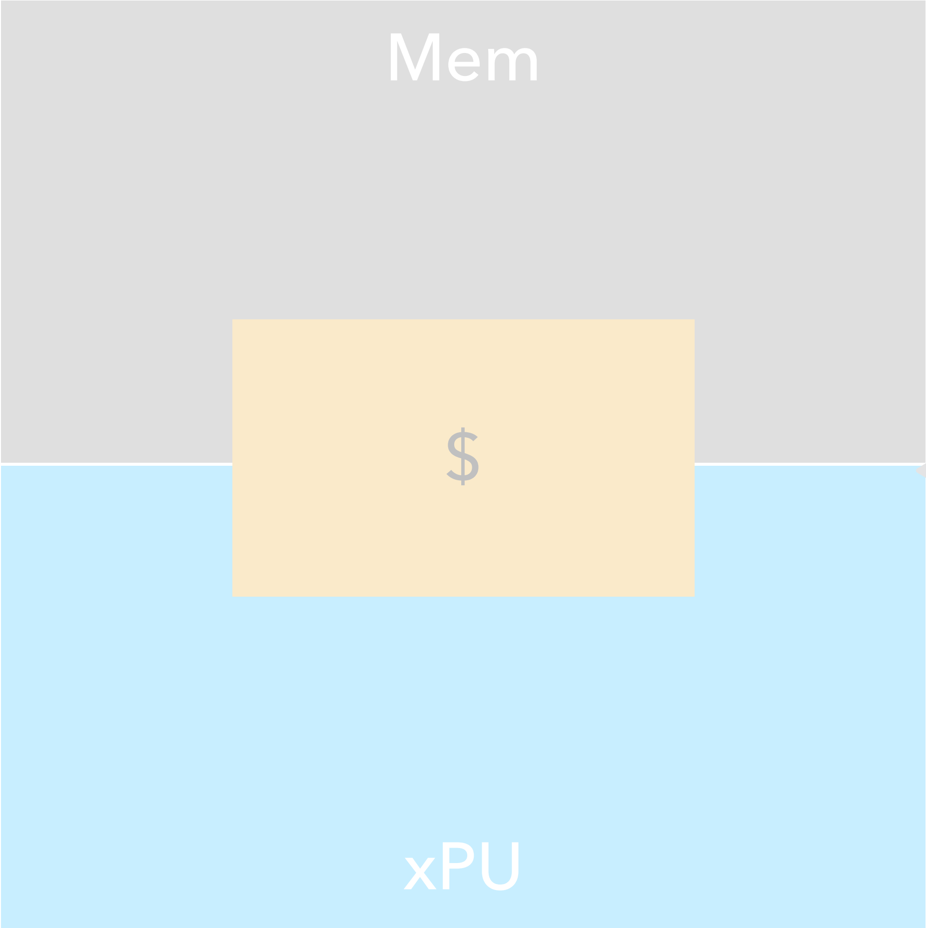
BF-2 yPUs (no host)



S. Karamati (GT), J. Young, R.V., et al. (2016) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

Hypothetical: Multi-SmartNIC

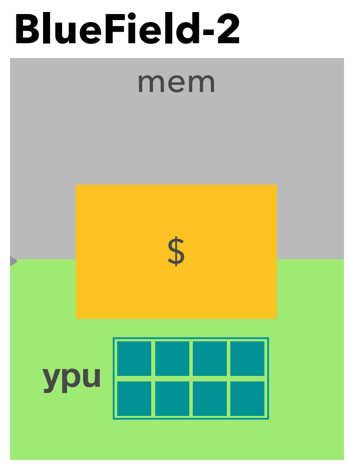
One host xPU (16 cores)



657 GF/s

76.8 GB/s

BF-2 yPUs (no host)



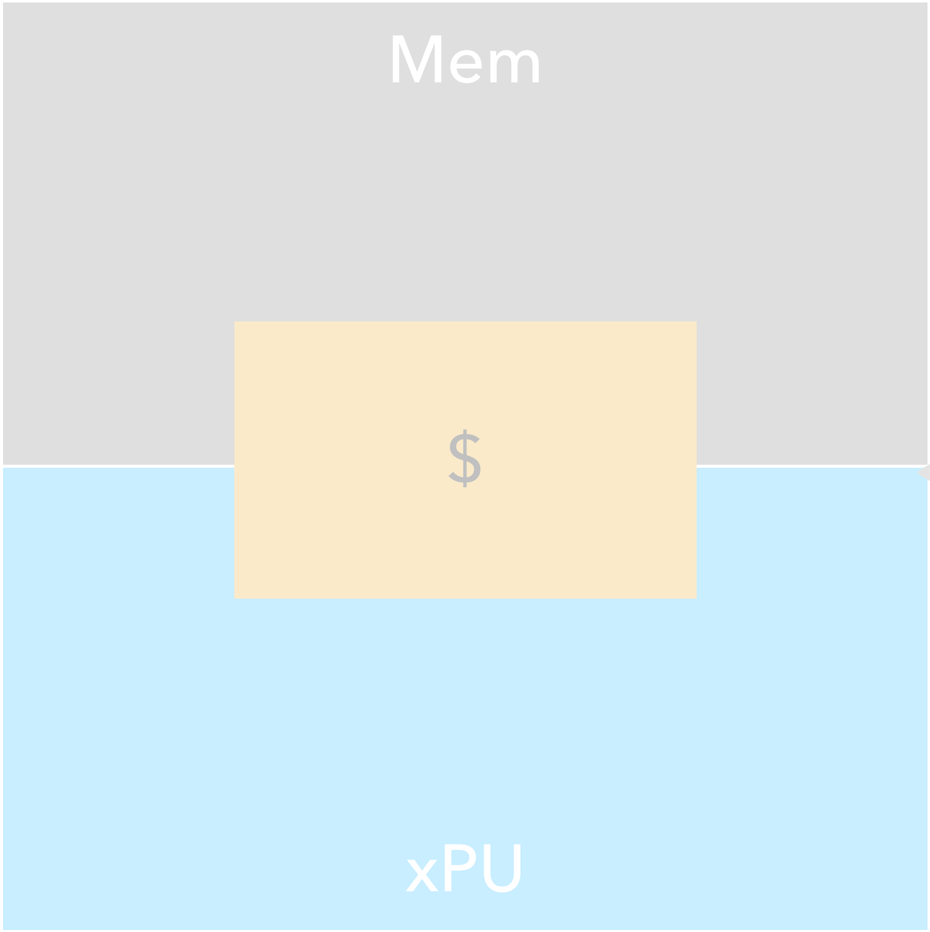
80 GF/s

25.6 GB/s

S. Karamati (GT), J. Young, R.V., et al. (2016) – “Smarter NICs for faster molecular dynamics: a case study.” doi:10.1109/IPDPS53621.2022.00063

Hypothetical: Multi-SmartNIC

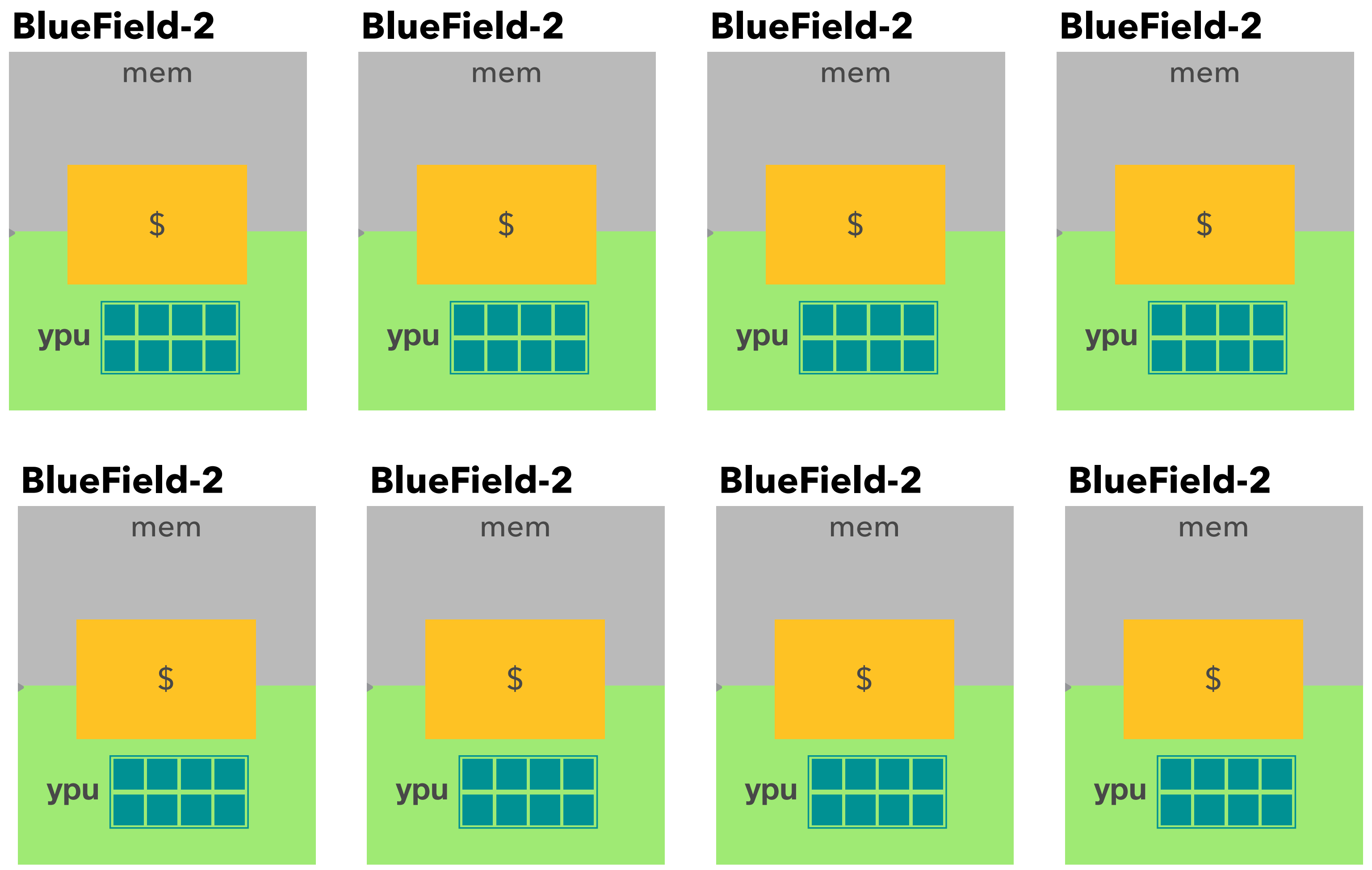
One host xPU (16 cores)



657 GF/s

76.8 GB/s

8 x BF-2 yPUs (no host)



640 GF/s

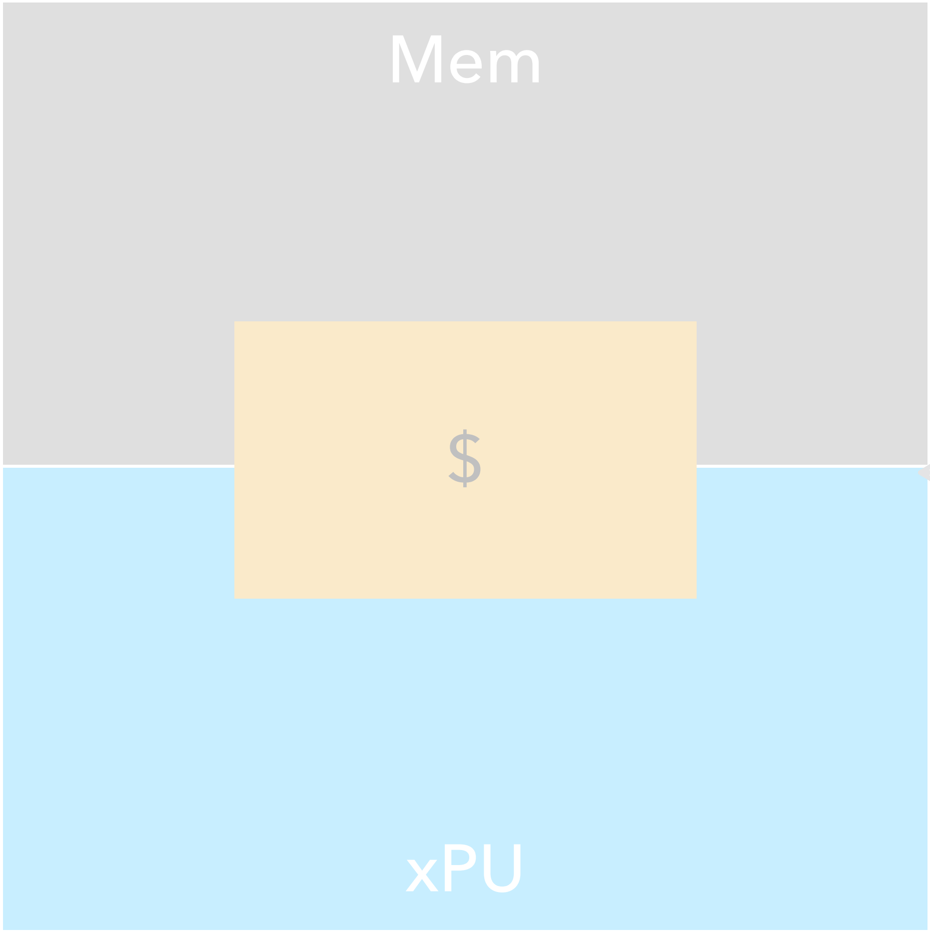
204 GB/s

(aggregate)

S. Karamati (GT), J. Young, R.V., et al. (2016) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

Hypothetical: Multi-SmartNIC

One host xPU (16 cores)



~ 8.5 F:B

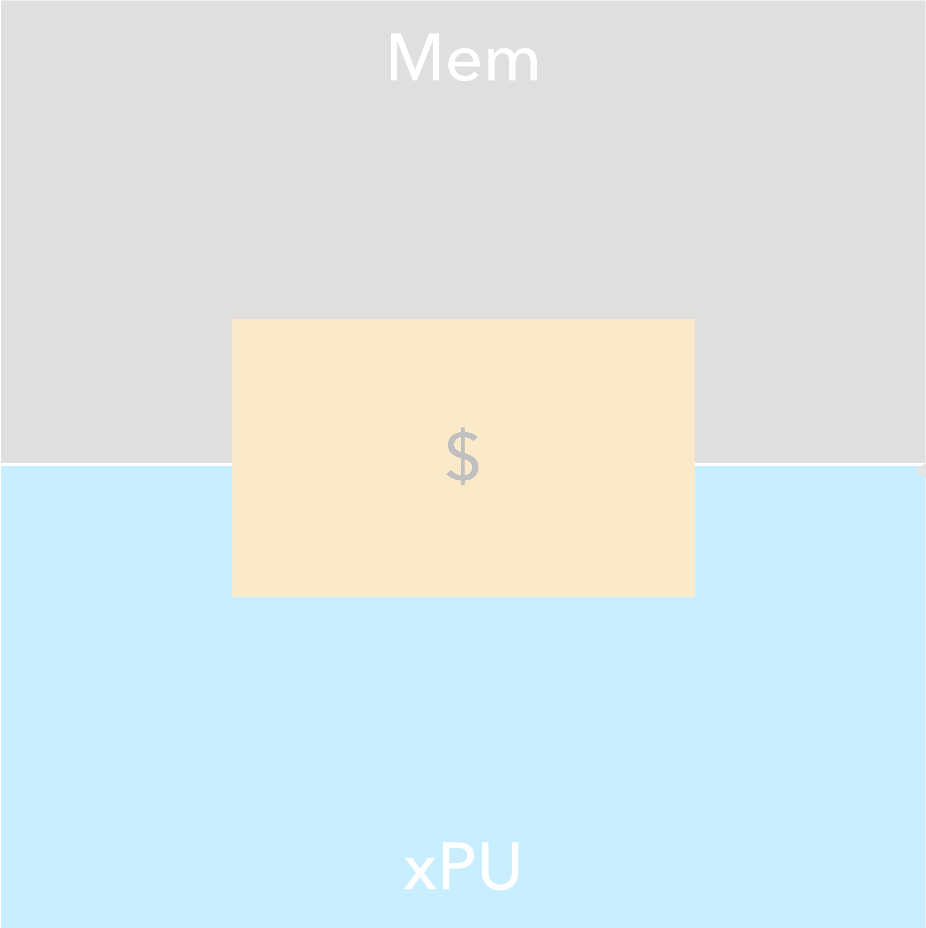
8 x BF-2 yPUs (no host)

~ 3.1 F:B

S. Karamati (GT), J. Young, R.V., et al. (2016) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

Hypothetical: Multi-SmartNIC

One host xPU (16 cores)



Time = "1"

using all cores

8 x BF-2 yPUs (no host)

Speedup ~ 1.7x

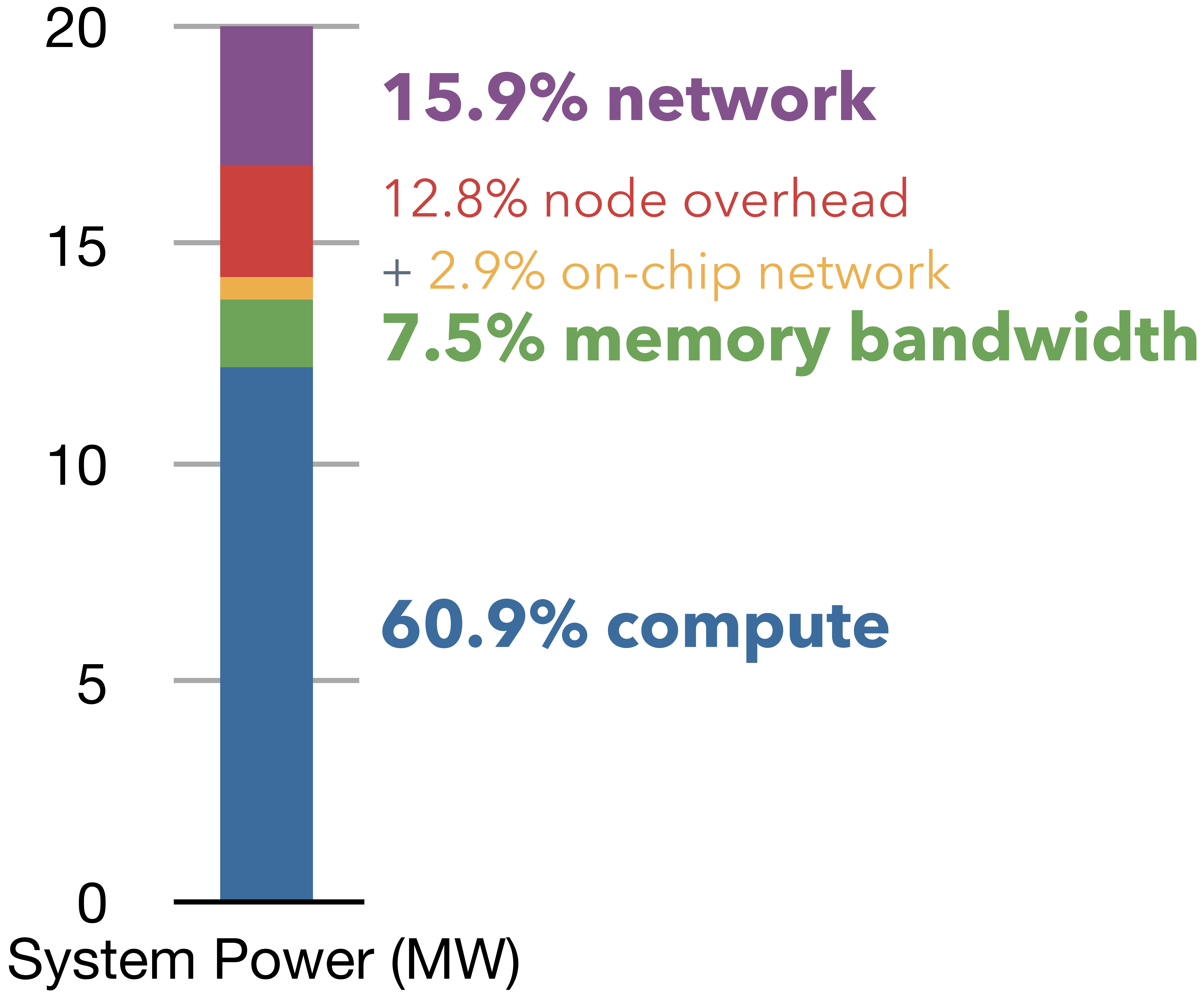
Real measurement on MiniMD!

S. Karamati (GT), J. Young, R.V., et al. (2016) – "Smarter NICs for faster molecular dynamics: a case study." doi:10.1109/IPDPS53621.2022.00063

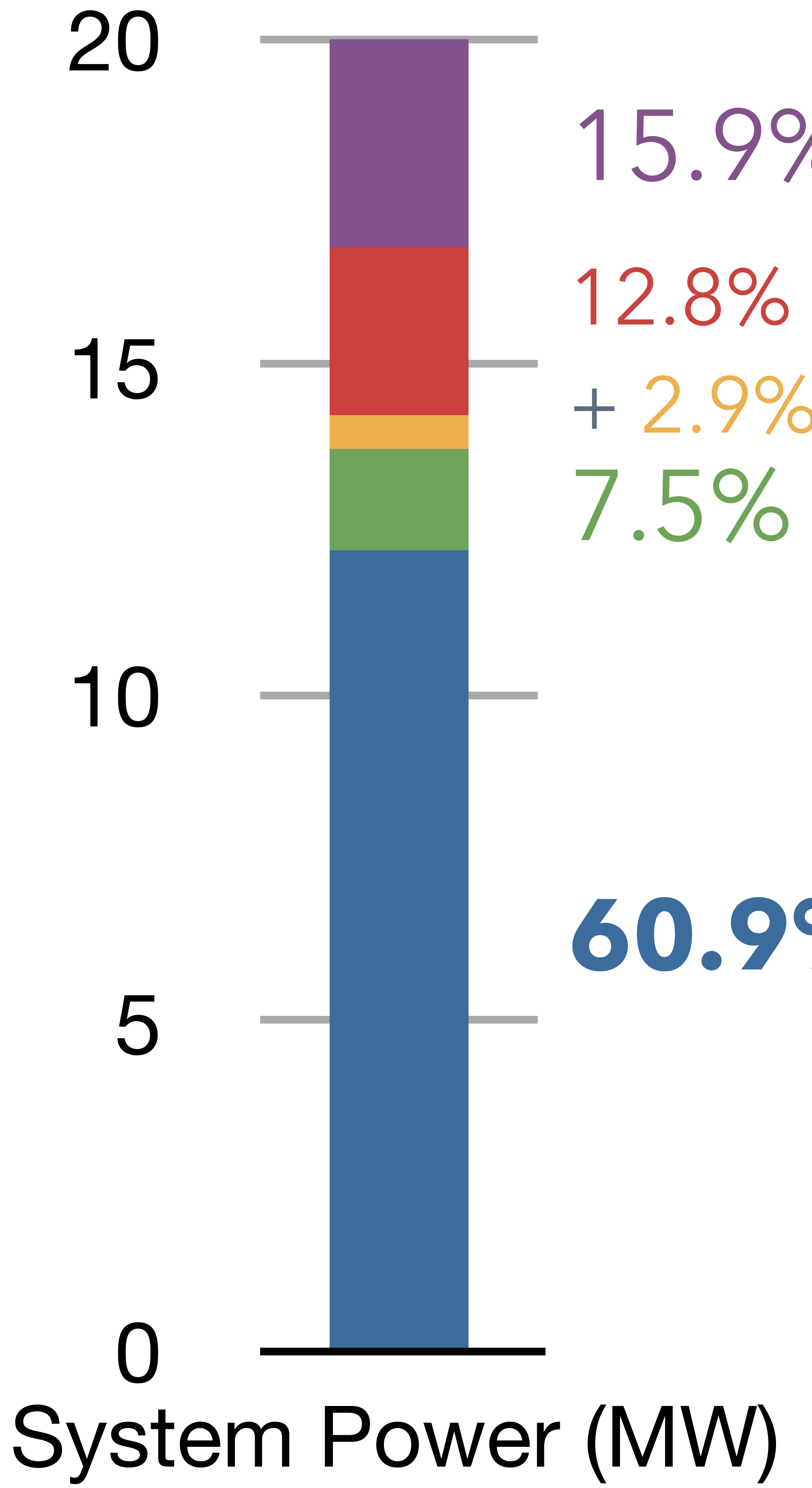
Power allocation for an "optimal" matrix multiply machine?

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



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

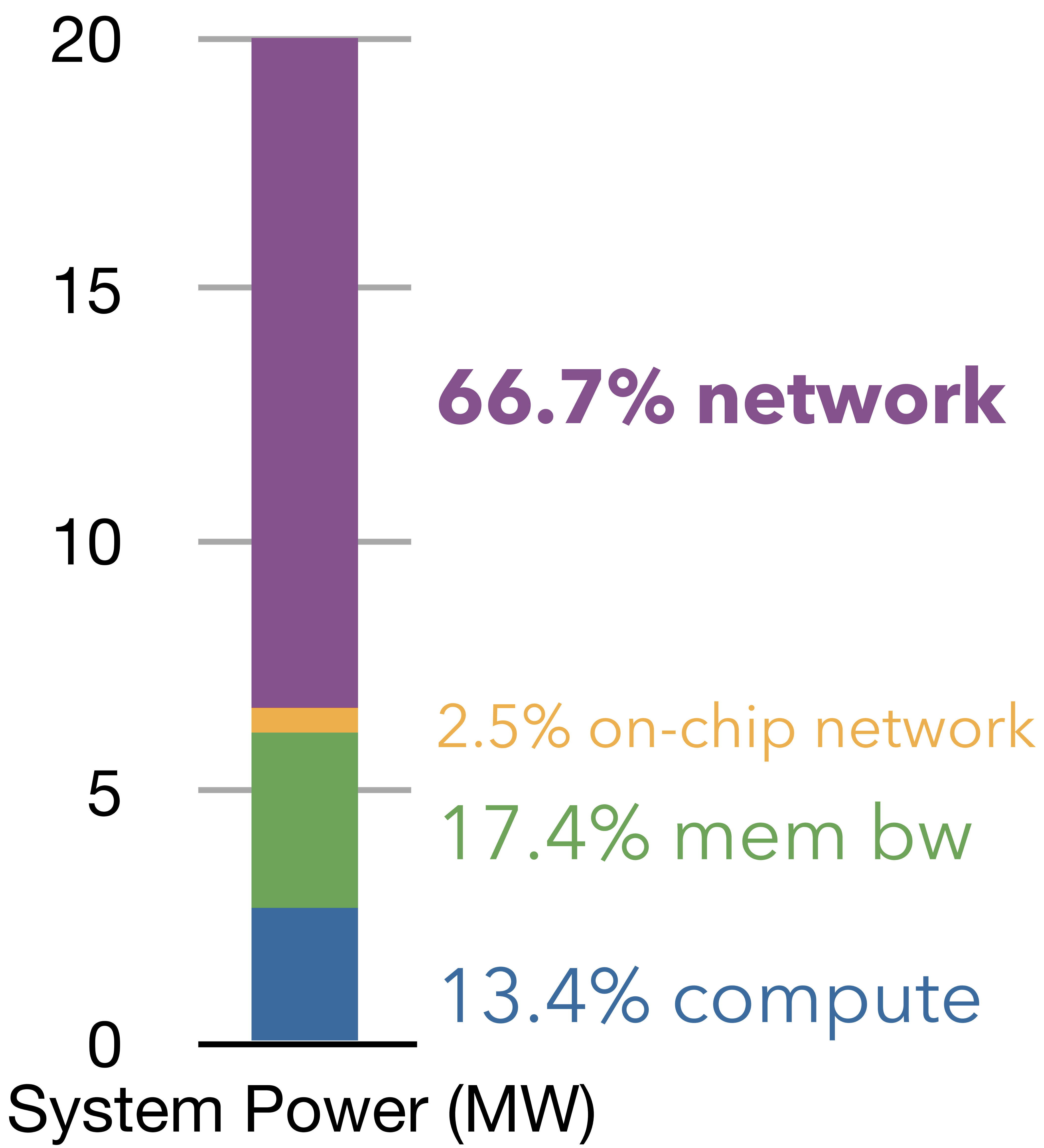
8% node overhead

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

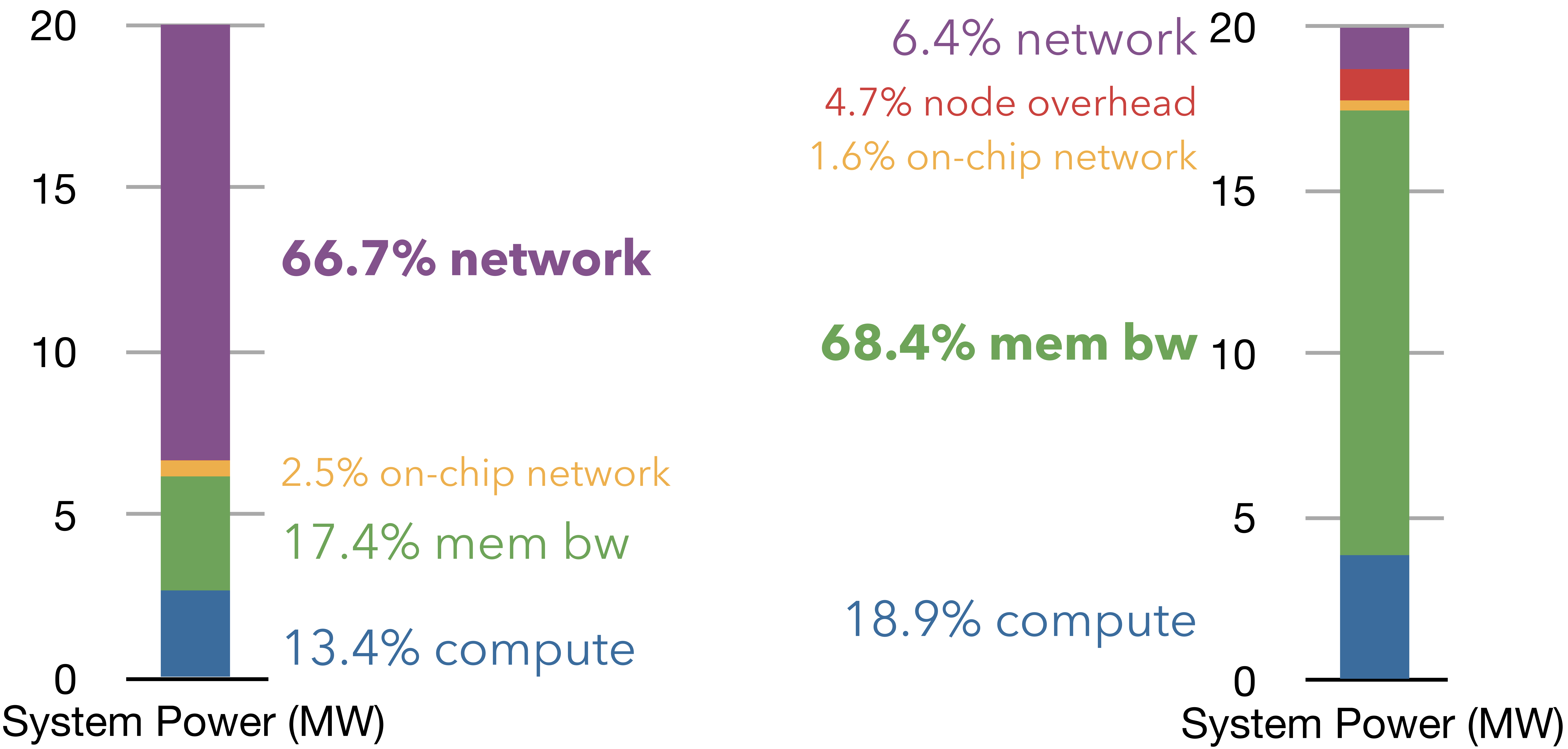
Power allocation for an “optimal” 3D FFT machine?

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

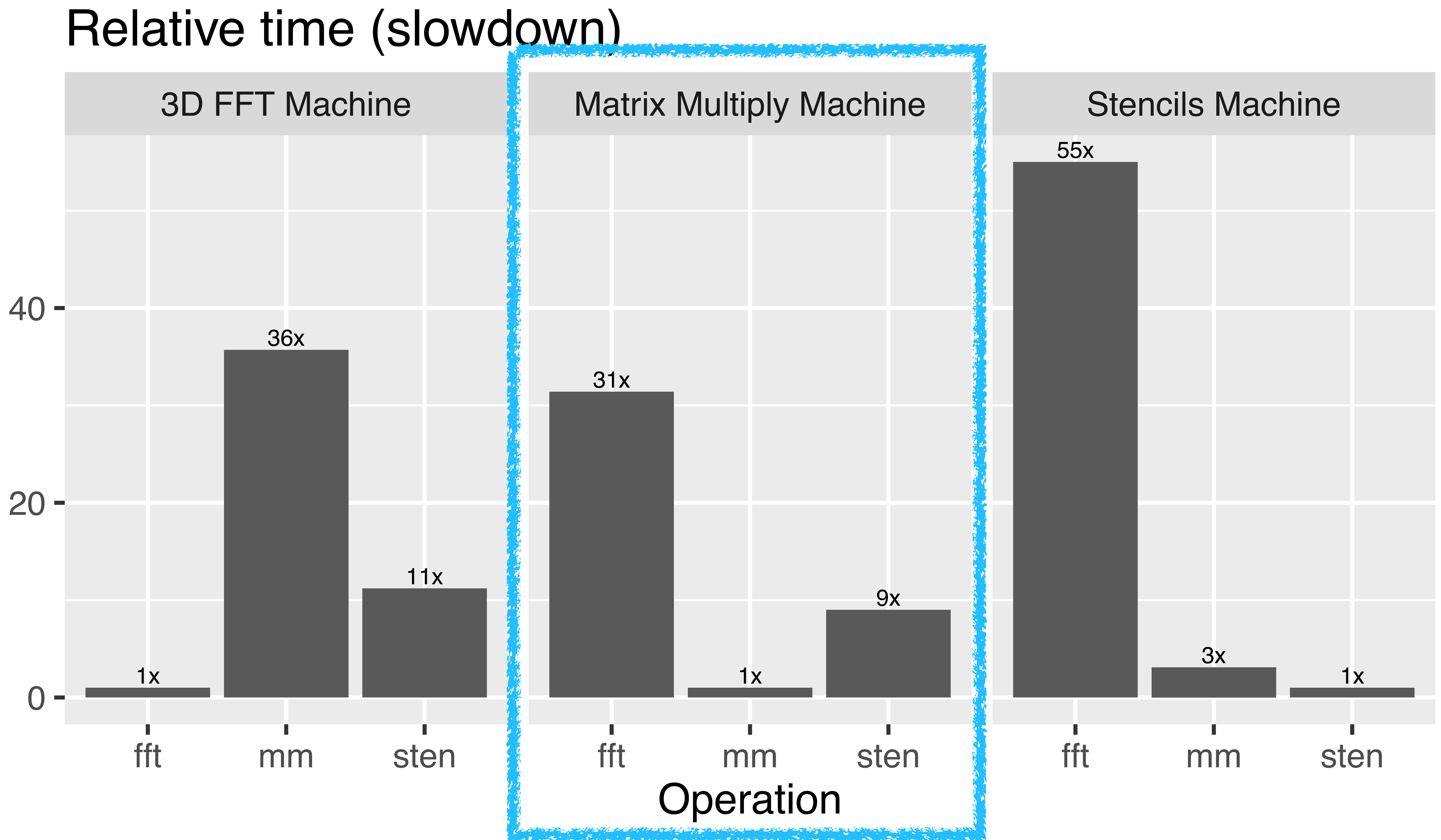
Power allocation for an "optimal" 3D FFT machine



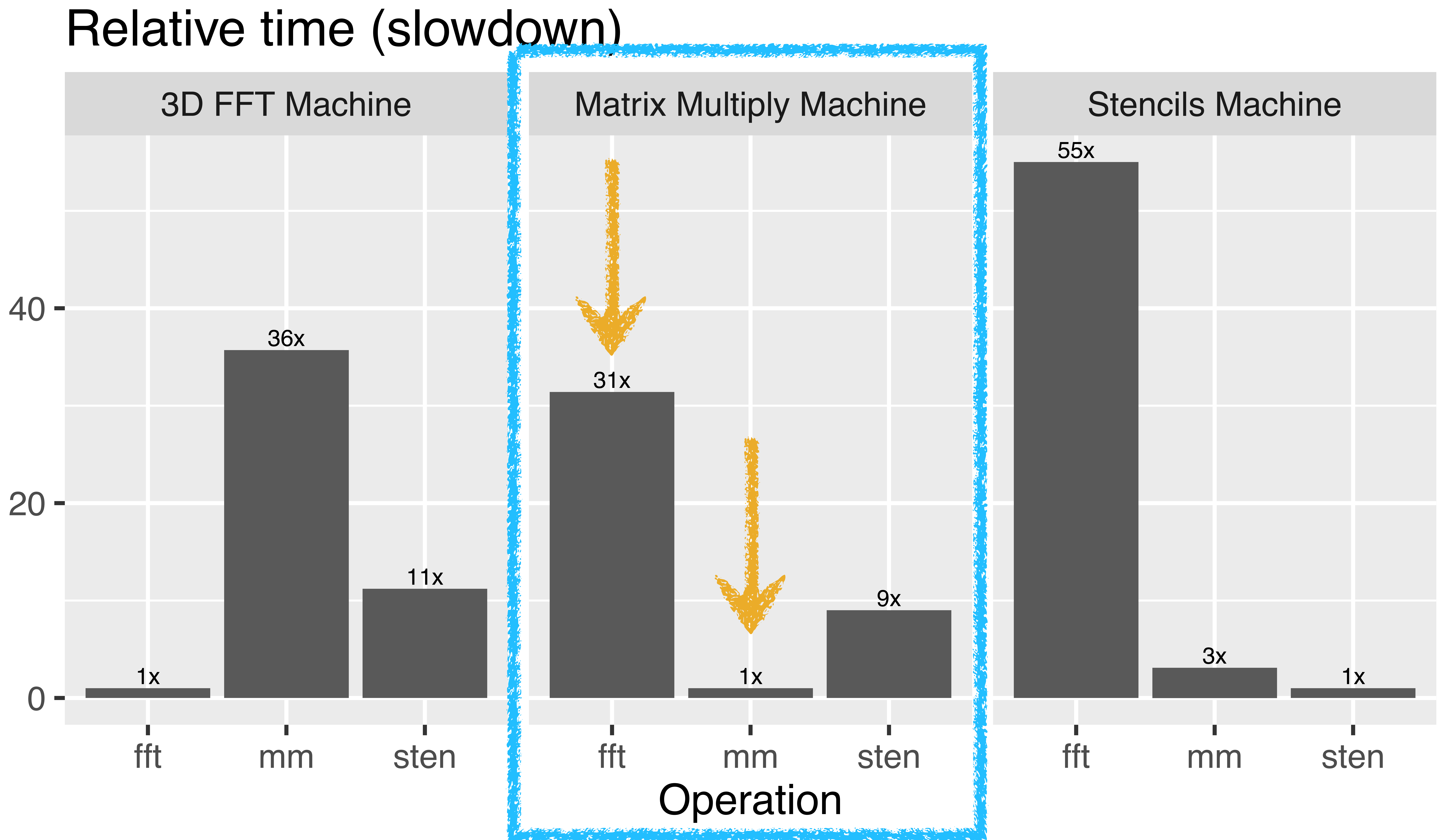
3D FFT vs. "Stencil" machines




Relative time (slowdown)
3D FFT Machine
Matrix Multiply Machine
Stencils Machine
1x
36x
11x
31x
1x
9x
55x
3x
1x
0
20
40
fft
mm
sten
fft
mm
sten
fft
mm
sten
Operation

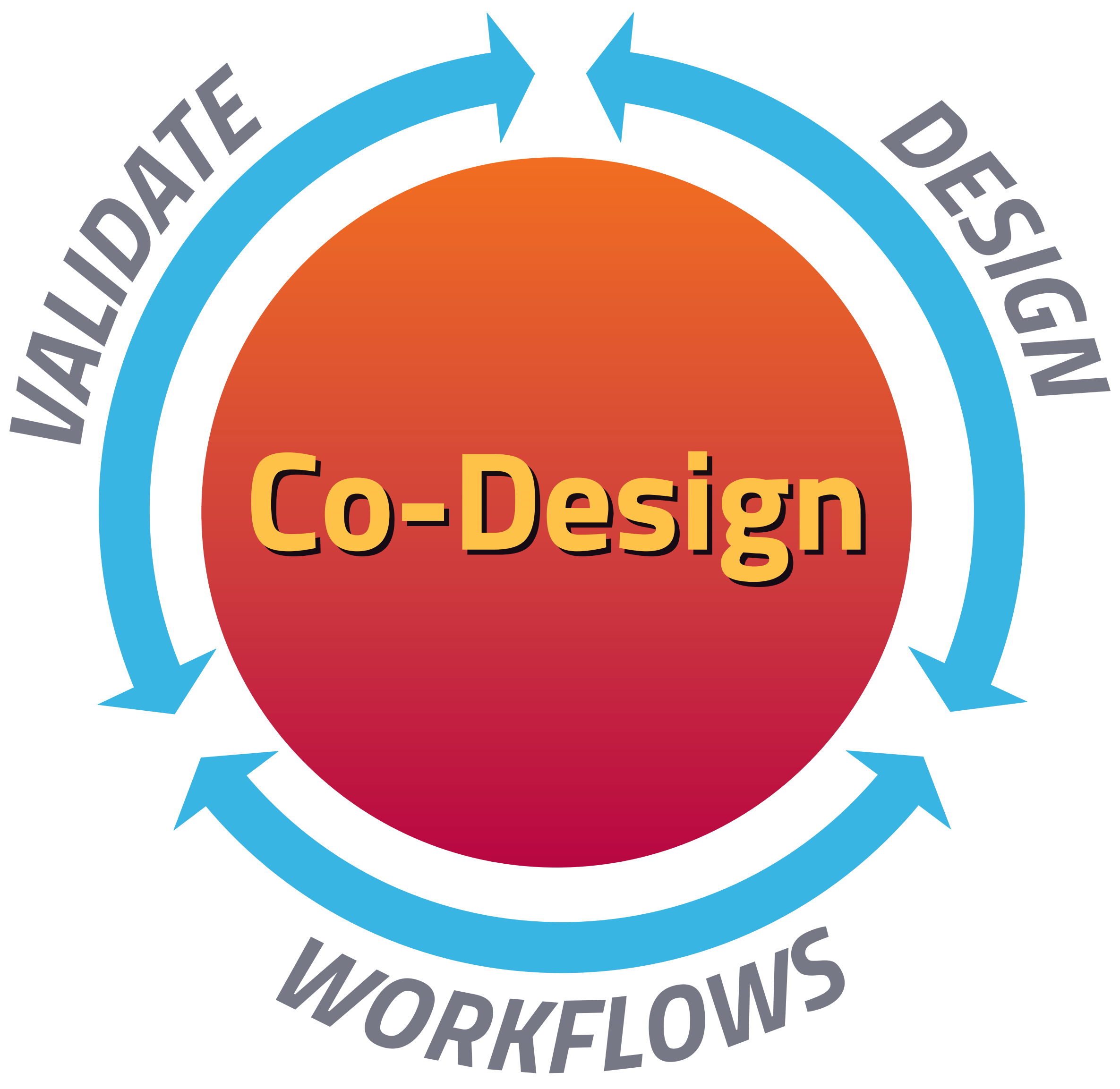


Relative time (slowdown)
3D FFT Machine
Matrix Multiply Machine
Stencils Machine
40
20
0
1x
36x
11x
31x
1x
9x
55x
3x
1x
fft
mm
sten
fft
mm
sten
fft
mm
sten
Operation


Relative time (slowdown)
3D FFT Machine
Matrix Multiply Machine
Stencils Machine
40
20
0
1x
36x
11x
31x
1x
9x
55x
3x
1x
fft
mm
sten
fft
mm
sten
fft
mm
sten
Operation

AGILE

ADVANCED GRAPHIC INTELLIGENCE LOGICAL COMPUTING ENVIRONMENT



Schematic of the AGILE Co-Design Process

of the applications. AGILE system designs must emphasize optimization of the fully integrated system rather than independent optimization of individual functionalities (e.g., memory, computation, or communication), and must not be constrained by existing component interfaces and protocols, legacy architectures, or current practices.

A fundamental rethinking of computer architectures that can revitalize performance growth trends in computing capabilities is long overdue. Currently, there is a renewed interest in developing specialized hardware components. However, this approach will not resolve the fundamental data movement challenges that restrict the historical performance growth trends. The AGILE program will seed a new generation

The AGILE BAA was released in November 2021 and the program is slated to run for three years.

TESTING AND EVALUATION PARTNERS

- Lawrence Berkeley National Laboratory
- Sandia National Laboratory
- Pacific Northwest National Laboratory

KEYWORDS

- Computer Architecture
- Data analytics
- Co-Design
- Data movement
- Modeling and simulation