

UNIVERSITÀ DEGLI STUDI DI TRIESTE

AREA SCIENCE PARK

Collecting Infrastructure-Level Measures for Energy Analysis in Modern Computing Infrastructures

Isac Pasianotto[1,2] • Stefano Cozzini[2] • Niccolò Tosato[1,2]

1 – University of Trieste • 2 – Area Science Park



SC26 hpc unites | Chicago

How much energy do 1,000 LLM tokens* cost to generate?

* Qwen3–VL-235B-Thinking – DGX node 8·Nvidia H100 – 4 GPUs used

How much energy do 1,000 LLM tokens* cost to generate?

0.56 Wh

Busy GPUs only

NVML counters – what is usually reported

* Qwen3–VL-235B-Thinking – DGX node 8·Nvidia H100 – 4 GPUs used

How much energy do 1,000 LLM tokens* cost to generate?

0.56 Wh → **1.27 Wh**

Busy GPUs only

NVML counters – what is usually reported

Whole node

+ idle GPUs + CPUs + RAM + fans + …

* Qwen3–VL-235B-Thinking – DGX node 8·Nvidia H100 – 4 GPUs used

How much energy do 1,000 LLM tokens* cost to generate?

0.56 Wh → **1.27 Wh** → **1.72 Wh**

Busy GPUs only

NVML counters – what is usually reported

Whole node

+ idle GPUs + CPUs + RAM + fans + …

Facility

× $PUE(t)$ measured during the runs

The facility pays 3.07× what the GPU counters report.

⋆ Qwen3–VL-235B-Thinking – DGX node 8·Nvidia H100 – 4 GPUs used

Energy is spread across the whole facility

A facility is more than its compute nodes

- Storage, network, services, and cooling draw power all the time
- Subsystems are coupled: a job on the compute nodes keeps storage servers and switches busy
- This shared energy is not attributed to any job

Energy is spread across the whole facility

A facility is more than its compute nodes

- Storage, network, services, and cooling draw power all the time
- Subsystems are coupled: a job on the compute nodes keeps storage servers and switches busy
- This shared energy is not attributed to any job

Most well-known public energy datasets are job-centric

- Per-job summary statistics (F-DATA [1])
- Or coarse, power-only traces (PM100 [2], 20 s)
- Or controlled single-application runs (C6EnPLS [3])

[1] F. Antici *et al.*, "F-DATA: A Fugaku workload dataset for job-centric predictive modelling in HPC systems"

[2] F. Antici *et al.*, "PM100: A job power consumption dataset of a large-scale production HPC system"

[3] M. Artioli *et al.*, "C6EnPLS: A high-performance computing job dataset for the analysis of linear solvers' power consumption"

We need to measure *the whole infrastructure*, with high-frequency, NTP-aligned time series.

Monitoring is not measuring!

Monitoring	Measuring
Best-effort scraping	Synchronized sampling, facility-wide
Sparse/adaptive intervals (minutes)	Fixed frequency (1+ Hz)
Occasional gaps are acceptable	Completeness is required and validated
Built for dashboards & alerting	**Built for quantitative analysis**

This work

This work

1

A reproducible measurement pipeline

Unmodified open-source components; fixed frequency, aligned and lossless; overhead and scaling quantified

This work

1

A reproducible measurement pipeline

Unmodified open-source components; fixed frequency, aligned and lossless; overhead and scaling quantified

2

A public, validated dataset

28 days · 1 Hz · ~70 nodes · Compute, storage, network, cloud and facility · 2.4×10^{10} rows

This work

1

A reproducible measurement pipeline

Unmodified open-source components; fixed frequency, aligned and lossless; overhead and scaling quantified

2

A public, validated dataset

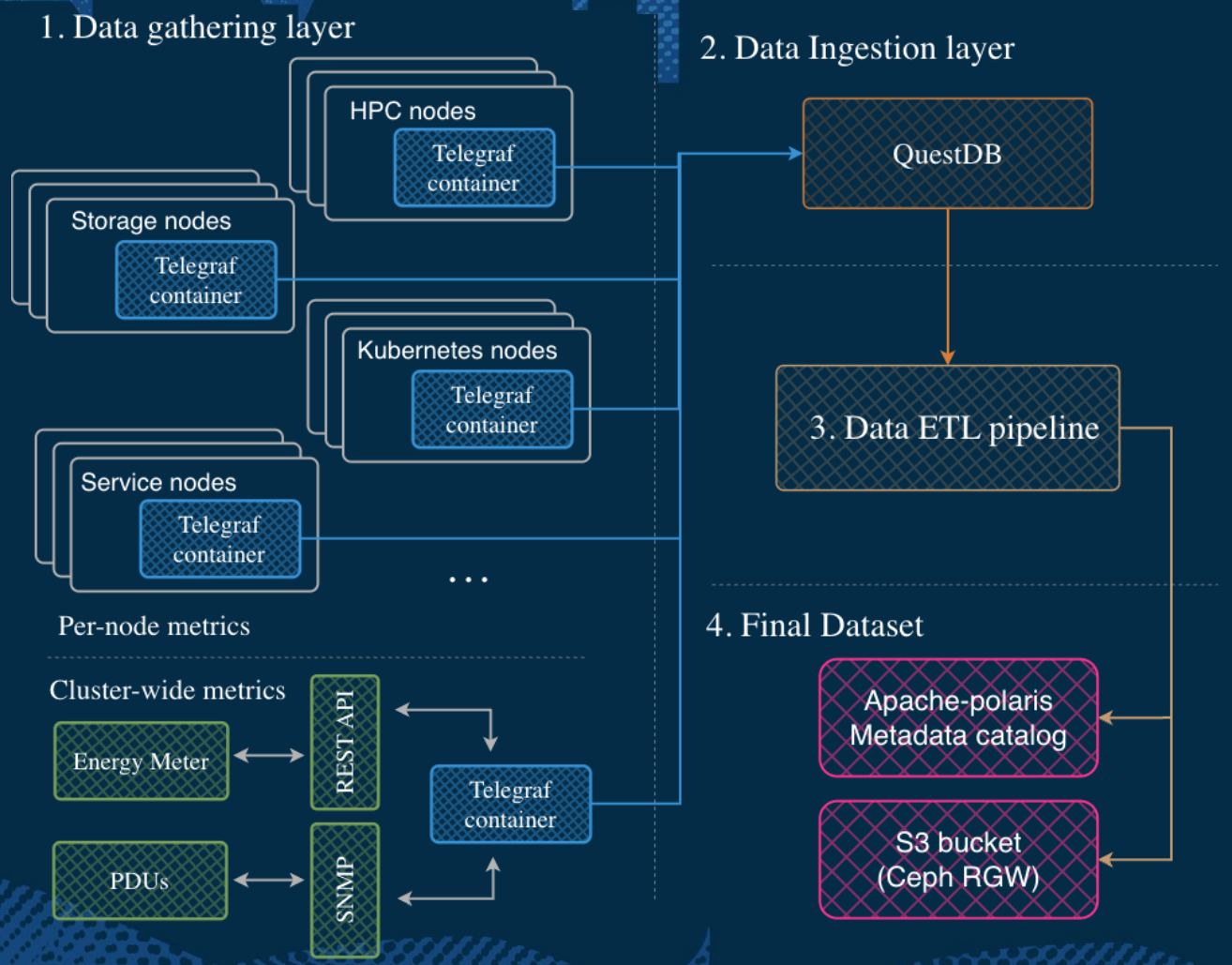
28 days · 1 Hz · ~70 nodes · Compute, storage, network, cloud and facility · 2.4×10^{10} rows

3

Two case studies

Energy efficacy of distributed I/O on Ceph · Per-token energy of LLM inference

A measurement pipeline from off-the-shelf parts



Gather

Telegraf container on every node, ad-hoc solution for cluster-wide metrics

No gaps

Agents push, buffer locally, and jitter their sends

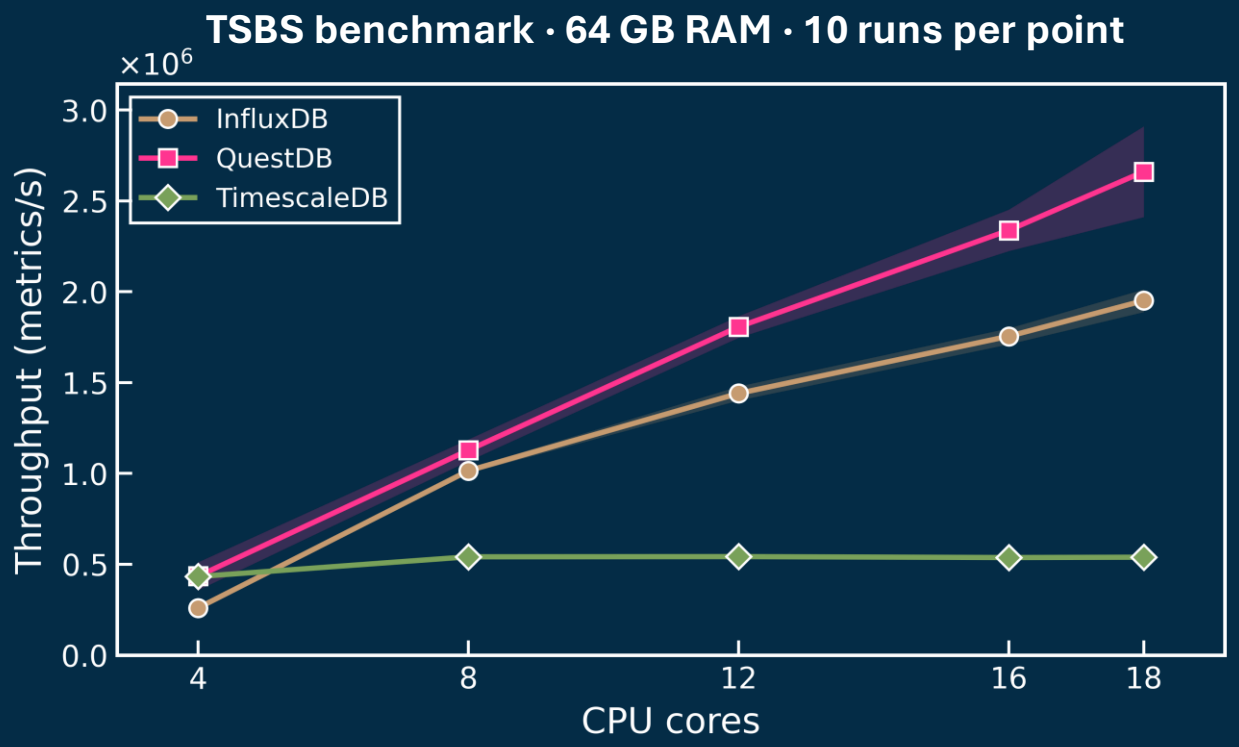
Ingest

QuestDB chosen for ingestion throughput

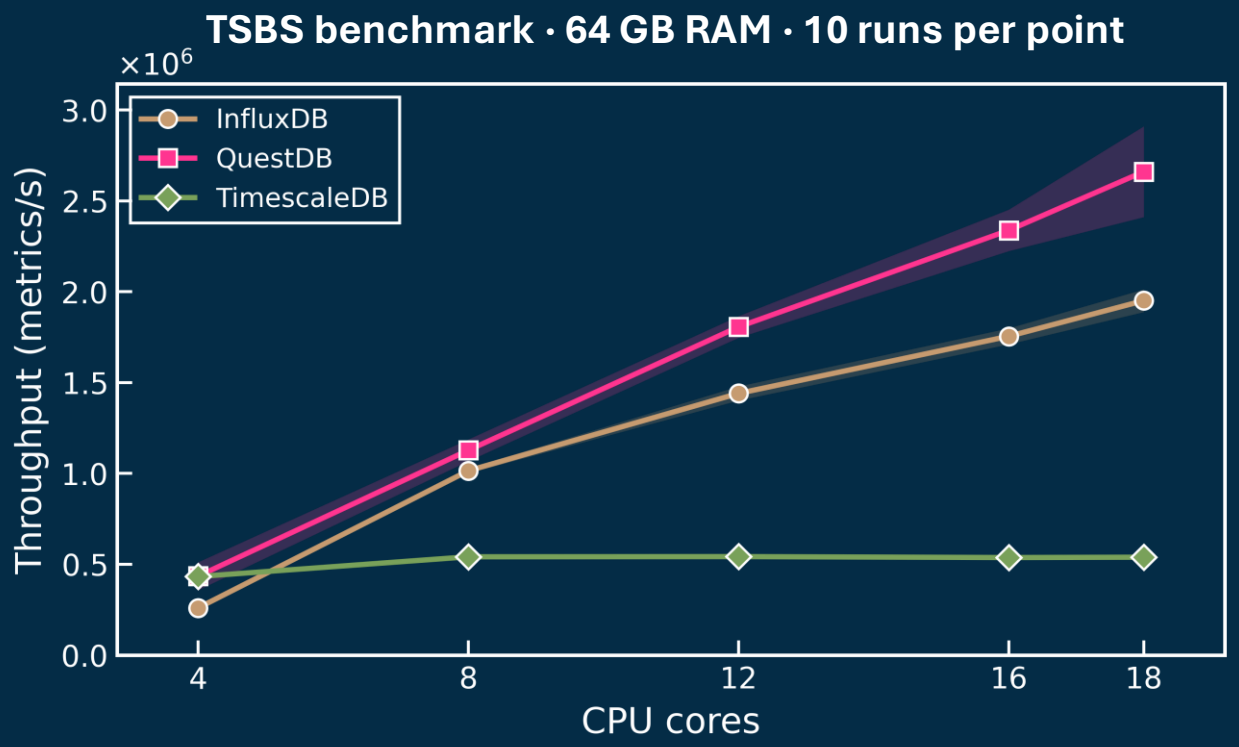
Archive

Daily Parquet files on S3+Apache Iceberg

Ingestion layer



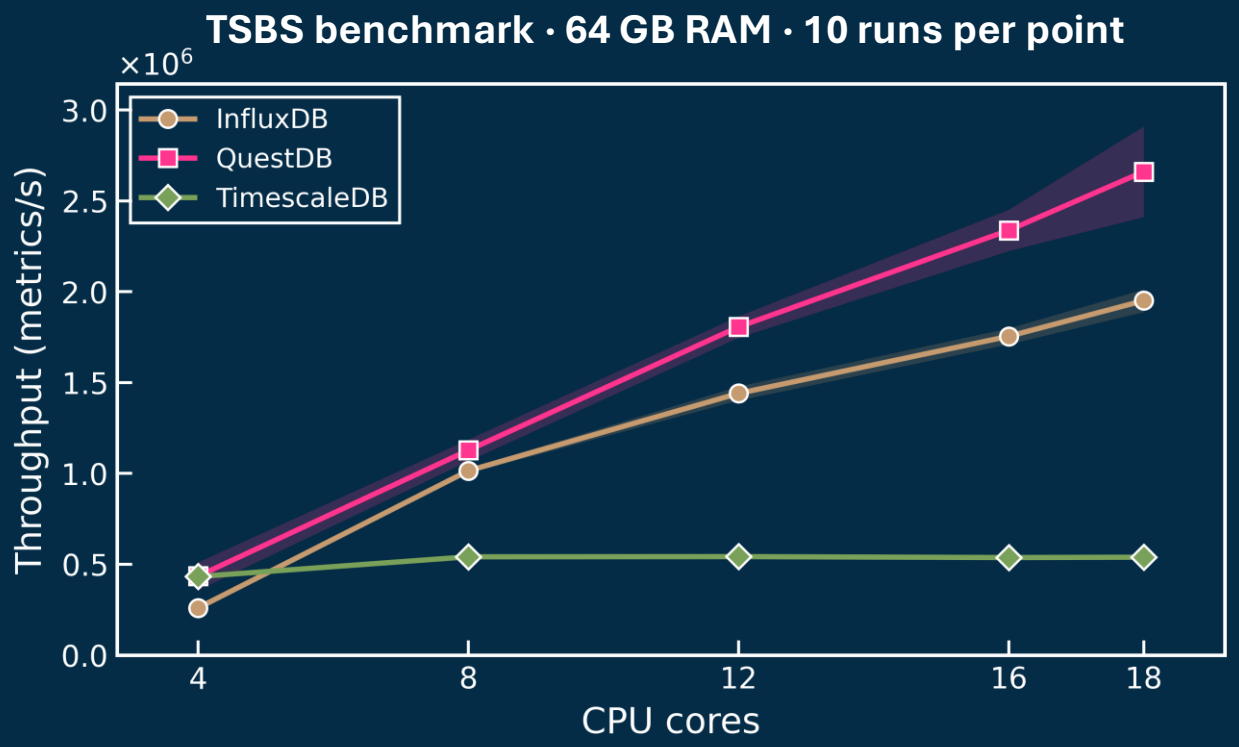
Ingestion layer



2.66M

metrics/s – QuestDB on 18 cores

Ingestion layer



2.66M

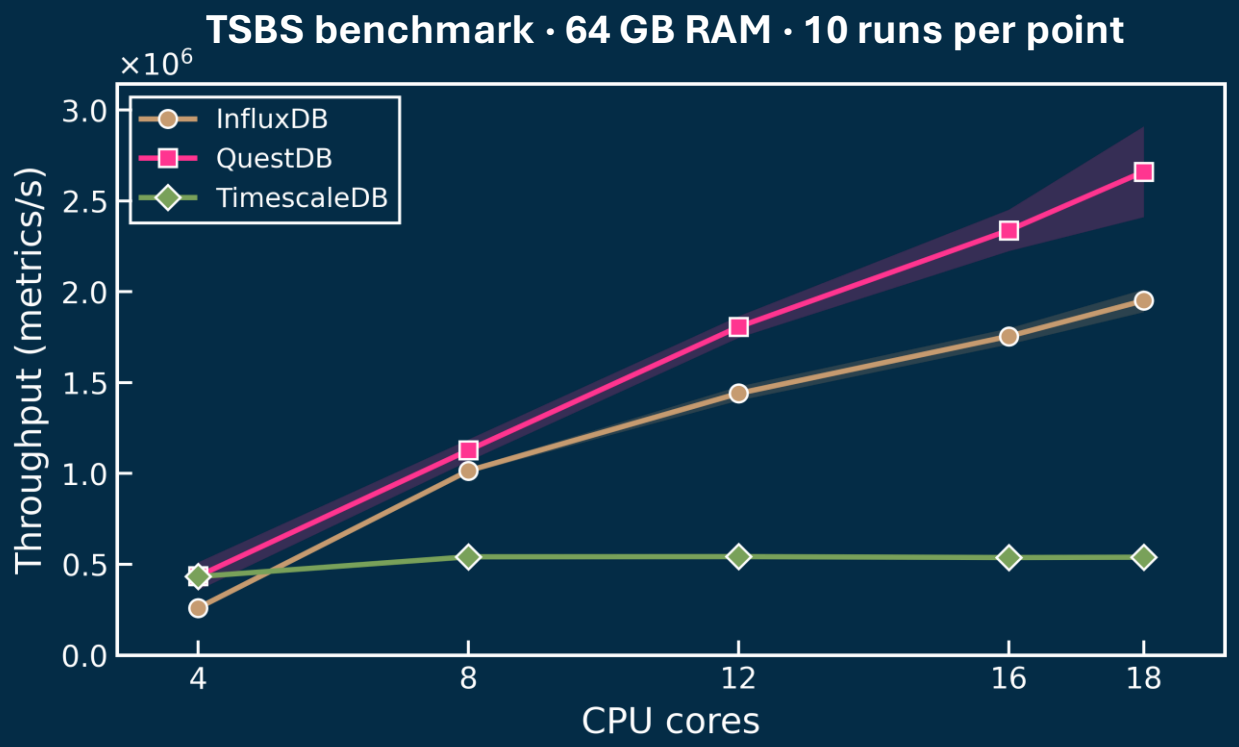
metrics/s – QuestDB on 18 cores

0.31M

metrics/s – our load: ~70 nodes at 1 Hz

Network ≈ 14KB/s per node

Ingestion layer



2.66M

metrics/s – QuestDB on 18 cores

0.31M

metrics/s – our load: ~70 nodes at 1 Hz

≥ 8.5×

headroom: ~ 1,000 nodes on 36 cores; shard hosts across instances to scale out

Network ≈ 14KB/s per node

28 days of 1 Hz data, from every layer

28 days of 1 Hz data, from every layer

28 days

continuous production operation

28 days of 1 Hz data, from every layer

28 days

continuous production operation

1 Hz

per node · PDU & energy meter: 30s

28 days of 1 Hz data, from every layer

28 days

continuous production operation

1 Hz

per node · PDU & energy meter: 30s

~70 nodes

HPC, Ceph storage, Kubernetes, Network

28 days of 1 Hz data, from every layer

28 days

continuous production operation

1 Hz

per node · PDU & energy meter: 30s

~70 nodes

HPC, Ceph storage, Kubernetes, Network

2.4×10^{10}

rows: 2.3TB → 366 GB Parquet

Open data, **CC BY 4.0**
doi.org/10.5281/zenodo.19468082



28 days of 1 Hz data, from every layer

28 days

continuous production operation

1 Hz

per node · PDU & energy meter: 30s

~70 nodes

HPC, Ceph storage, Kubernetes, Network

2.4×10^{10}

rows: 2.3TB → 366 GB Parquet

What's inside

- Kernel counters – cpu, mem, diskio, infiniband
- Hardware sensors – RAPL, IPMI power & sensors, NVML
- Facility – PDU sockets, energy meters
- Context – Slurm jobs, LLM API usage

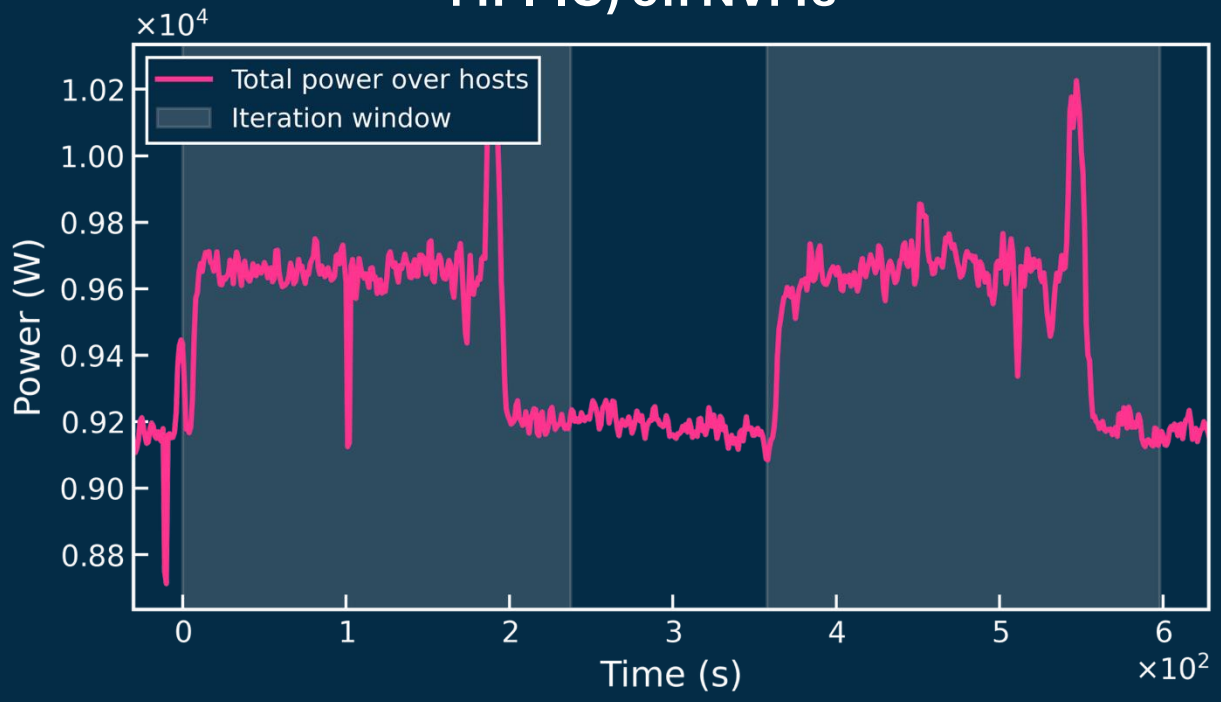
Open data, **CC BY 4.0**

doi.org/10.5281/zenodo.19468082

“Built for research and quantitative analysis” …

What does I/O cost on a production Ceph cluster?

Total power of 23 Ceph storage servers: two IOR iterations of sequential write (4MB, MPI-IO) on NVMe

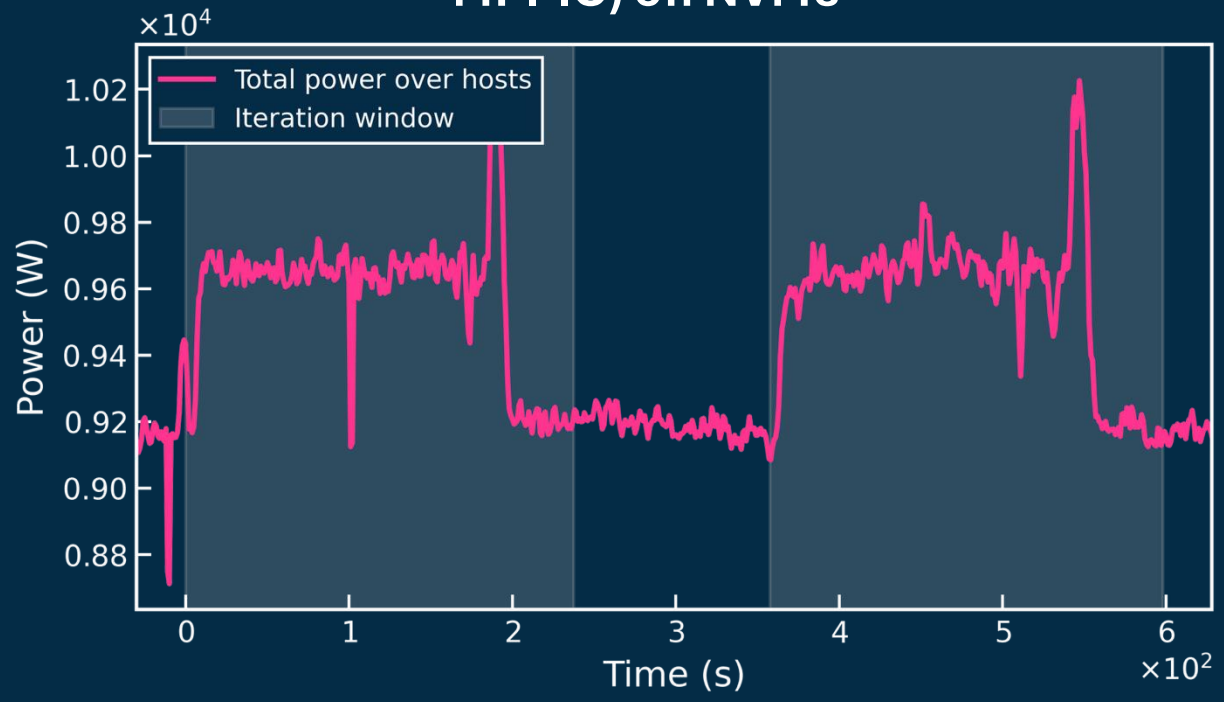


Setup:

- IOR launched on 2 128-cores compute nodes on CephFS
- HDD pool → 6+2 erasure coding
- NVMe pool → replica 2
- seq / rand · 4 KB / 4 MB · POSIX / MPI-IO · 10 × 3-minutes runs

What does I/O cost on a production Ceph cluster?

Total power of 23 Ceph storage servers: two IOR iterations of sequential write (4MB, MPI-IO) on NVMe



Setup:

- IOR launched on 2 128-cores compute nodes on CephFS
- HDD pool → 6+2 erasure coding
- NVMe pool → replica 2
- seq / rand · 4 KB / 4 MB · POSIX / MPI-IO · 10 × 3-minutes runs

+4.9%

9.2 kW → 9.65 kW under load:
the energy of I/O is mostly the cost of availability

Write 1TB: a microwave … or a home for one month

Write 1TB: a microwave … or a home for one month

3.6x

More data with the same energy

1.49 MB/J NVMe → 0.41 MB/J HDDs

Sequential write · 4MB block size · POSIX backend

Write 1TB: a microwave … or a home for one month

3.6x

More data with the same energy

1.49 MB/J NVMe → 0.41 MB/J HDDs

Sequential write · 4MB block size · POSIX backend

0.19 kWh

To write 1TB in the best case

≈ a microwave running for 10 minutes

Sequential write · 4MB block size · POSIX backend · NVMe

Write 1TB: a microwave … or a home for one month

3.6x

More data with the same energy

1.49 MB/J NVMe → 0.41 MB/J HDDs

Sequential write · 4MB block size · POSIX backend

0.19 kWh

To write 1TB in the best case

≈ a microwave running for 10 minutes

Sequential write · 4MB block size · POSIX backend · NVMe

~278 kWh

To write 1TB in the worst case

≈ 26 days of electricity for an average home in Europe

Sequential write · 4KB block size · POSIX backend · HDDs

Per-token energy attribution for an LLM service

- shared, multi-tenant LLM service
- 1 node: 8× Nvidia H100 · 2 × Intel Xeon 8480+
- 3 models, distributed across 2/4/8 GPUs
- Locust: 20 users × 200 prompts × 15 min per run

$$E_{sys} = E_{CPU} + E_{GPU}^{busy} + E_{GPU}^{idle} + E_{other}$$

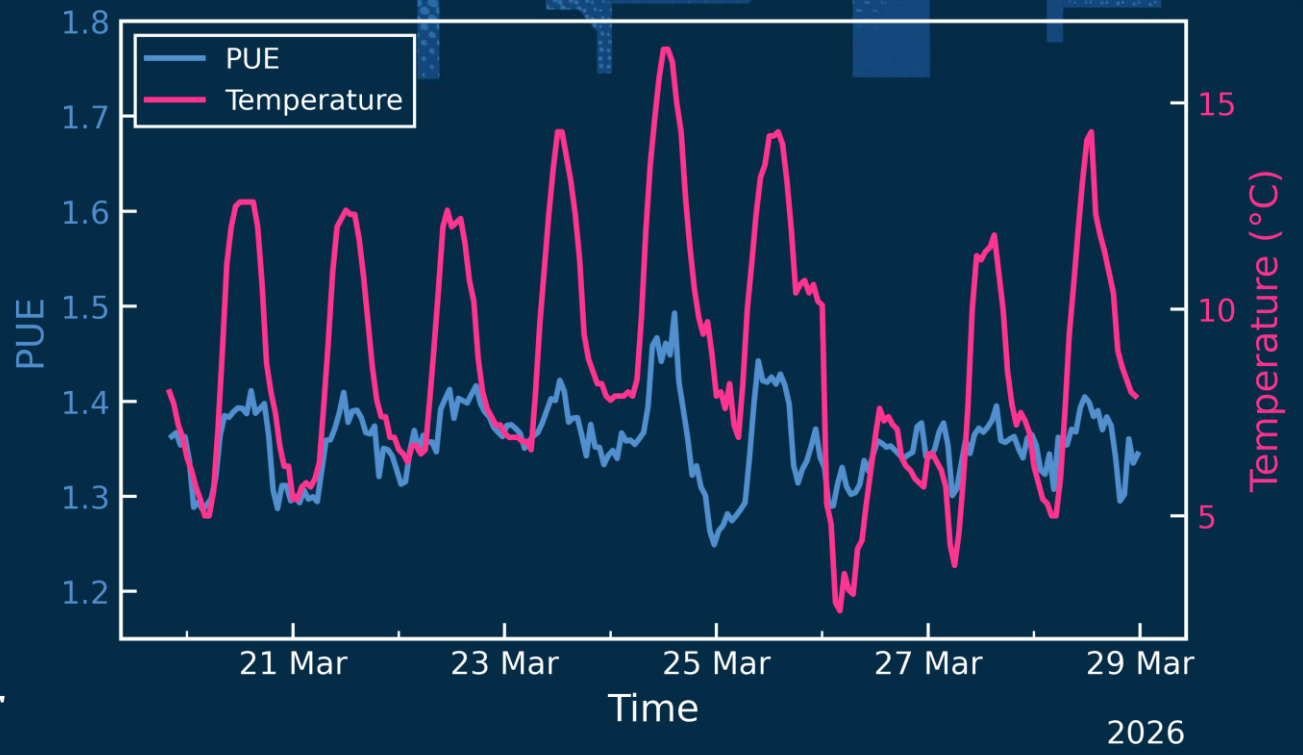
$$E_{fac} = E_{sys} \cdot PUE$$

Per-token energy attribution for an LLM service

- shared, multi-tenant LLM service
- 1 node: 8× Nvidia H100 · 2 × Intel Xeon 8480+
- 3 models, distributed across 2/4/8 GPUs
- Locust: 20 users × 200 prompts × 15 min per run

$$E_{sys} = E_{CPU} + E_{GPU}^{busy} + E_{GPU}^{idle} + E_{other}$$

$$E_{fac} = E_{sys} \cdot PUE\ (t)$$

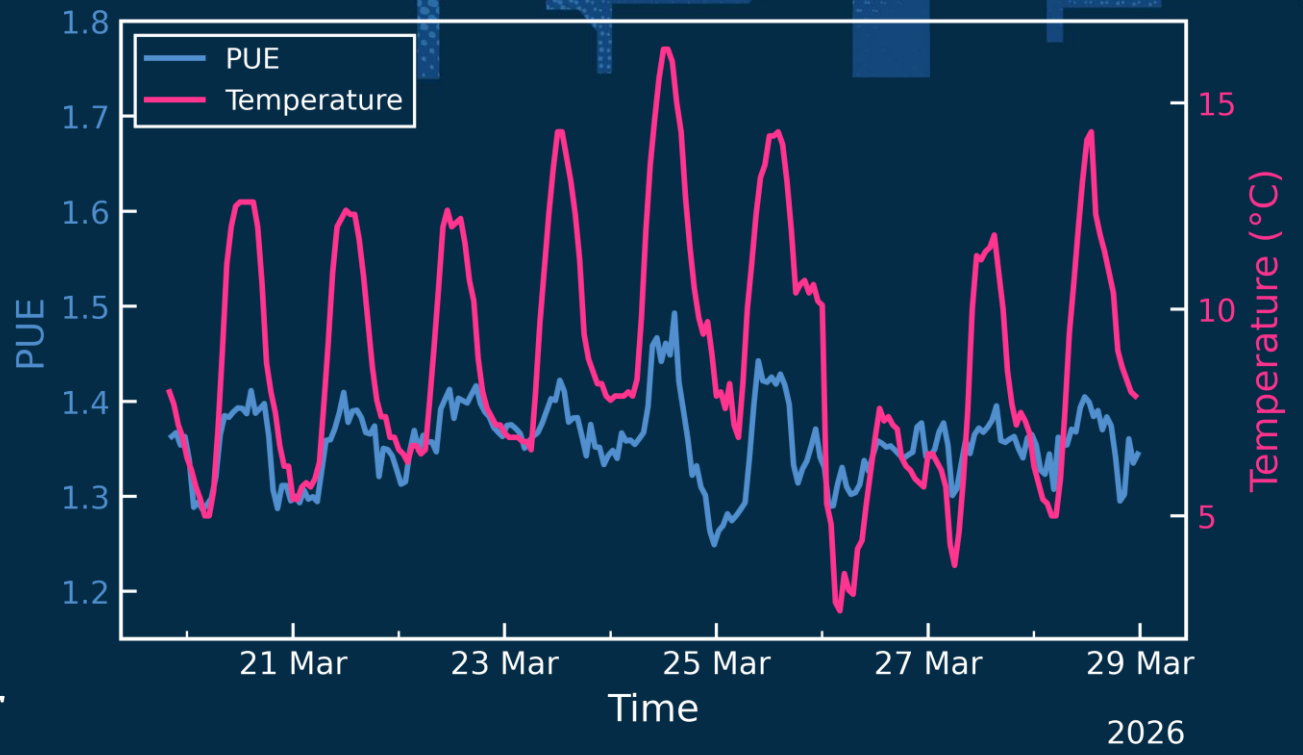


Per-token energy attribution for an LLM service

- shared, multi-tenant LLM service
- 1 node: 8× Nvidia H100 · 2 × Intel Xeon 8480+
- 3 models, distributed across 2/4/8 GPUs
- Locust: 20 users × 200 prompts × 15 min per run

$$E_{sys} = E_{CPU} + E_{GPU}^{busy} + E_{GPU}^{idle} + E_{other}$$

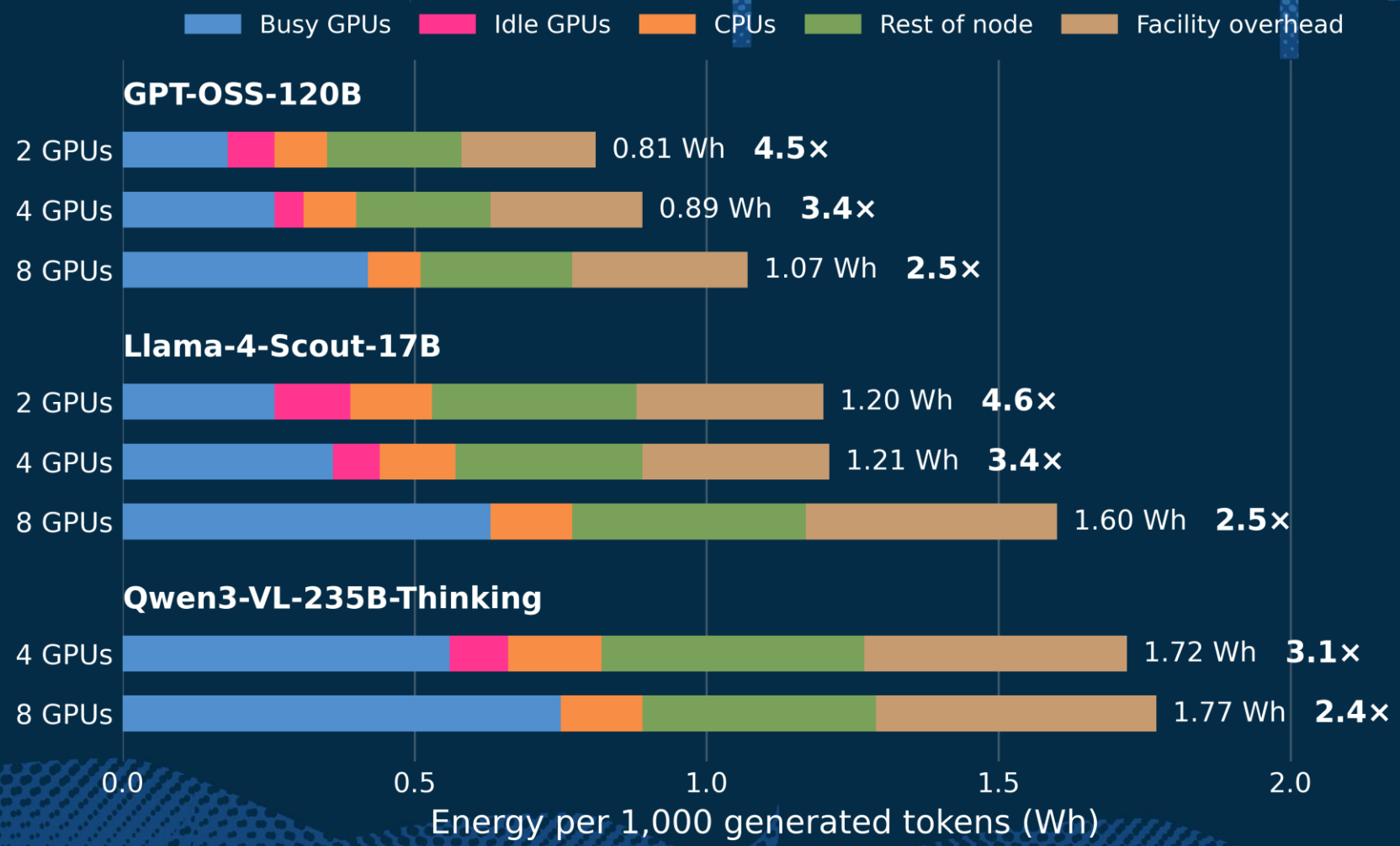
$$E_{fac} = E_{sys} \cdot PUE\ (t)$$



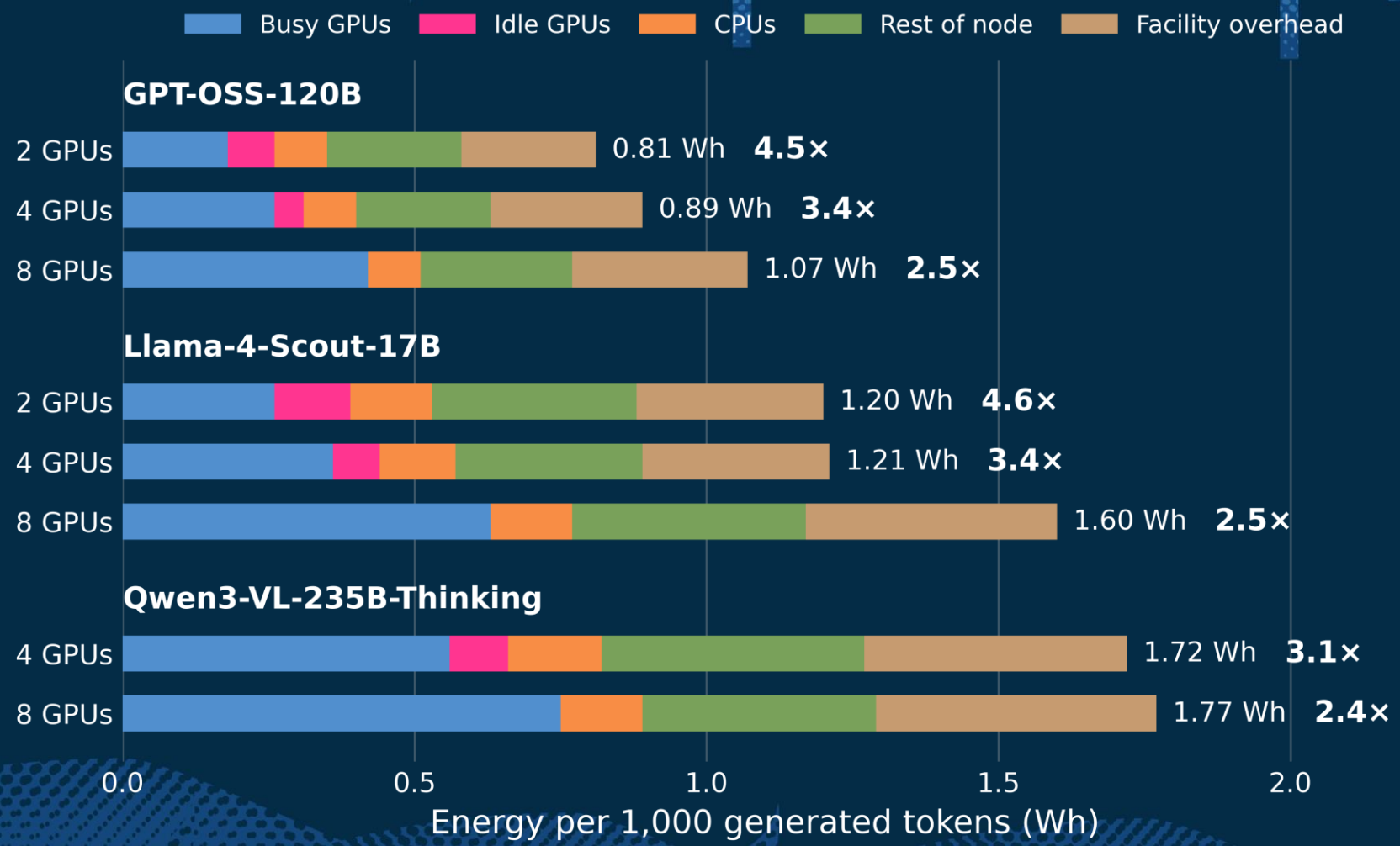
1.25 – 1.60

The same job can cost up to ~28% more depending on when it runs!

GPUs are less than half of the true cost!



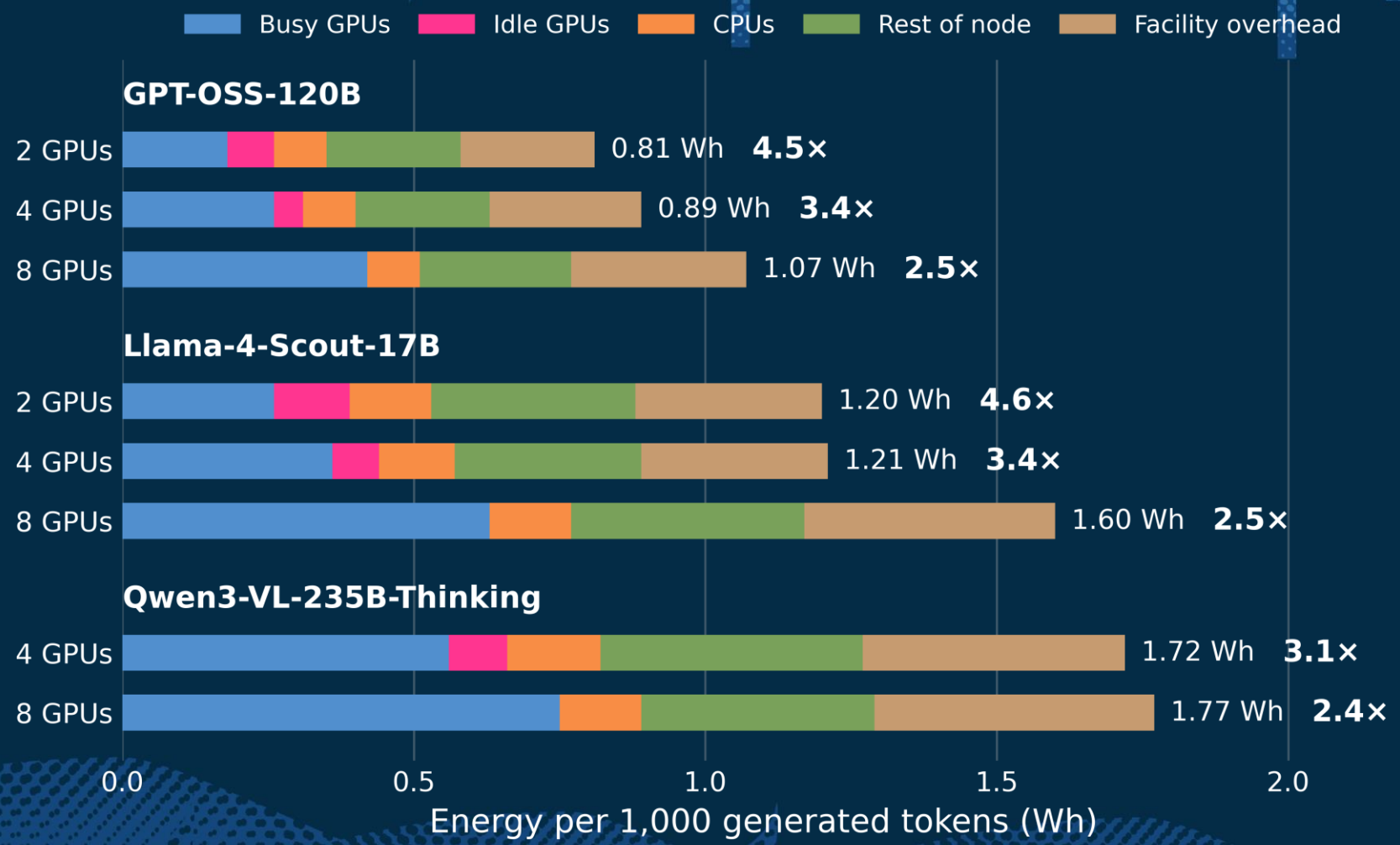
GPUs are less than half of the true cost!



2.4 – 4.6 ×

GPU accounting underestimates the facility cost

GPUs are less than half of the true cost!



2.4 – 4.6 ×

GPU accounting underestimates the facility cost

2→8 GPUs

GPT-OSS: same 1,500 tokens/s
Llama-4: same 1,000 tokens/s

Qwen3: 800 → 960 tokens/s

Takeaways

1. Measure, don’t just monitor

Fixed-frequency, aligned, lossless, validated data that covers the entire cluster

2. Infrastructure-level data changes the numbers

Component-centric analysis can be misleading

3. Everything is open

Dataset and deployment are public:
replicate the setup with off-the-shelf
open-source tools and publish comparable
data

github.com/IsacPasianotto/infrastructure-level-data-collection

Thank you for your attention!

✉ isac.pasianotto@phd.units.it