Through Time, We See.

Leveraging High-Precision Time Synchronization for Enhanced Profiling and Debugging in Distributed AI Systems

Gal Korcia, Wojtek Wasko

When the job becomes the cluster

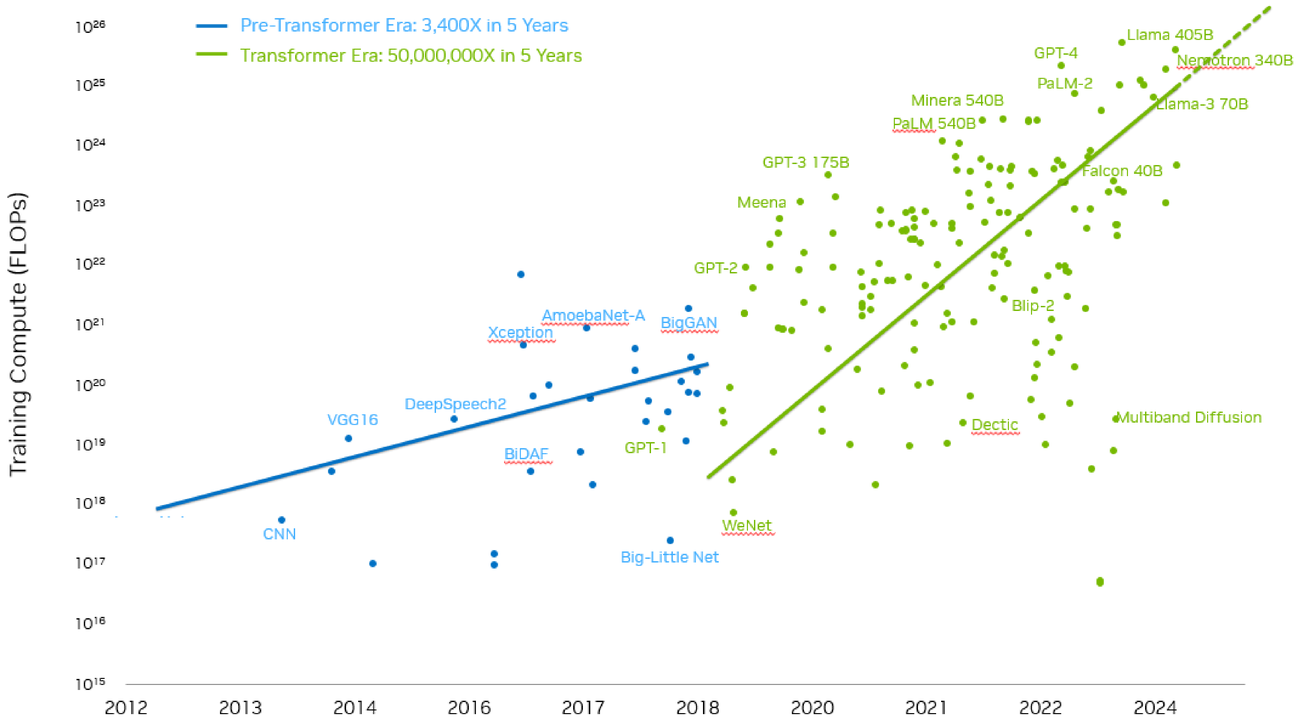
The window for sync issues grew from hours to weeks.

- **HPC, even at leadership scale (Titan, Summit, Frontier)**
  - Big tightly-coupled jobs run for hours-to-days
  - Cluster time-multiplexed across diverse workloads

- **Frontier AI training (Llama 3.1)**
  - One job
  - ~30.8M hours across ~16,000 GPUs working together

Sources: Lim et al., IEEE Cluster 2019 (Titan); OLCF 2024 Operational Assessment Report (Frontier); Meta Llama 3.1 Model Card (Jul 2024).

Exploding Energy Usage In Data Centers



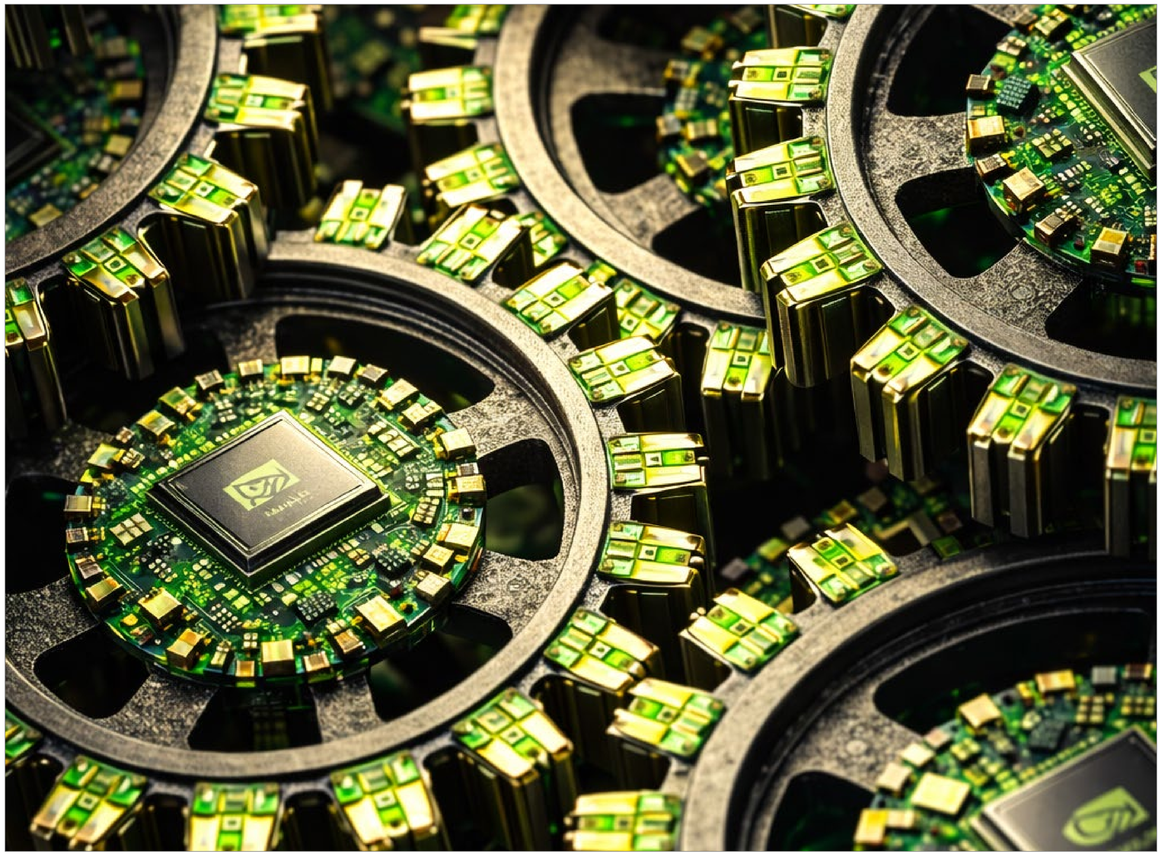
At this scale, efficiency is existential.

The lever is stragglers.

Model sizes: Github AI Memory Wall, Arxiv papers | Data Center Electricity Usage: IEA report

Mechanics of AI training

distilled down



GPU POV:

- calculate my part (compute)
- merge results (collective communication)
- repeat

1000’s times per second.

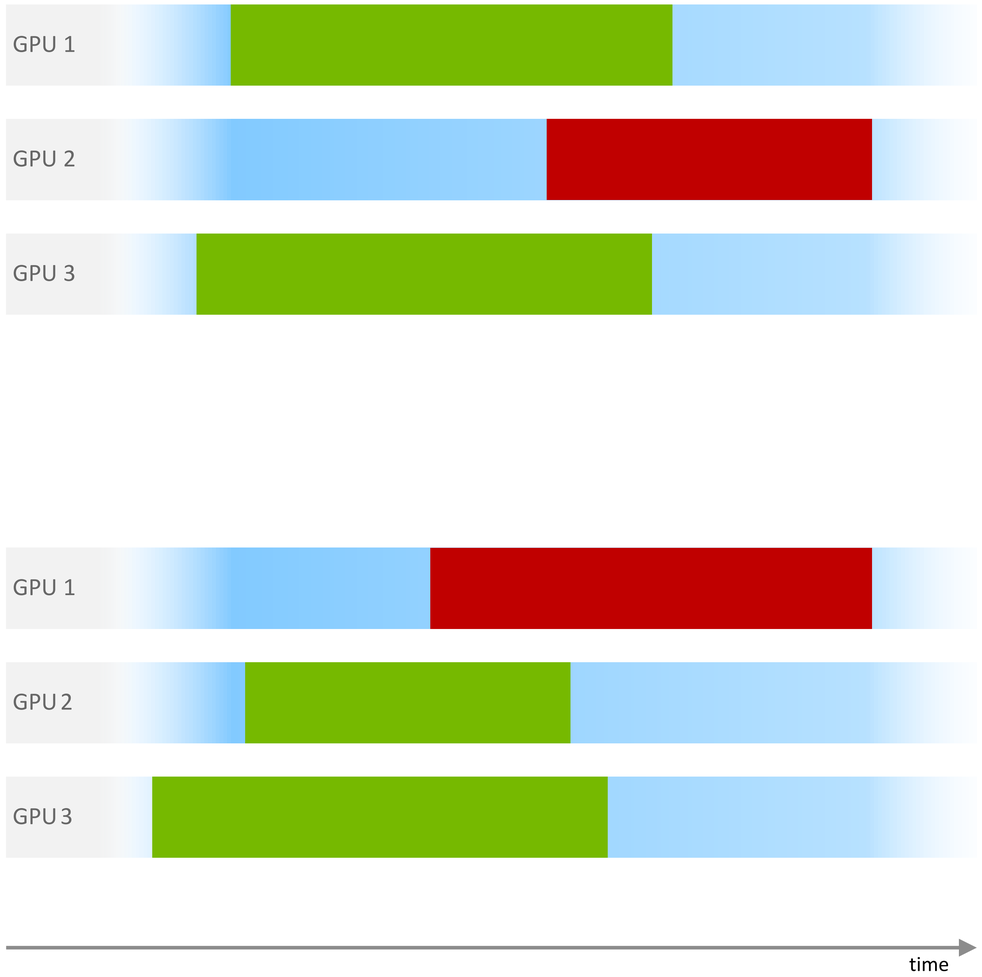
Collectives expose delays.

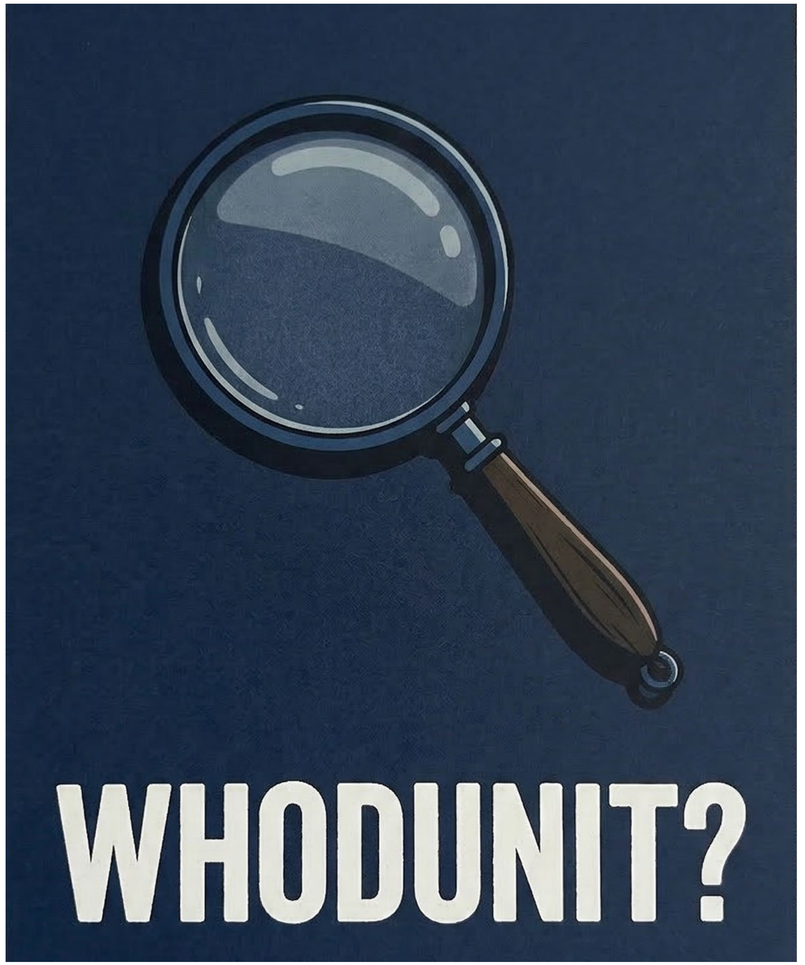
Delays accumulate and propagate.

One slow node slows down many others.

A common time axis

Same data, different views

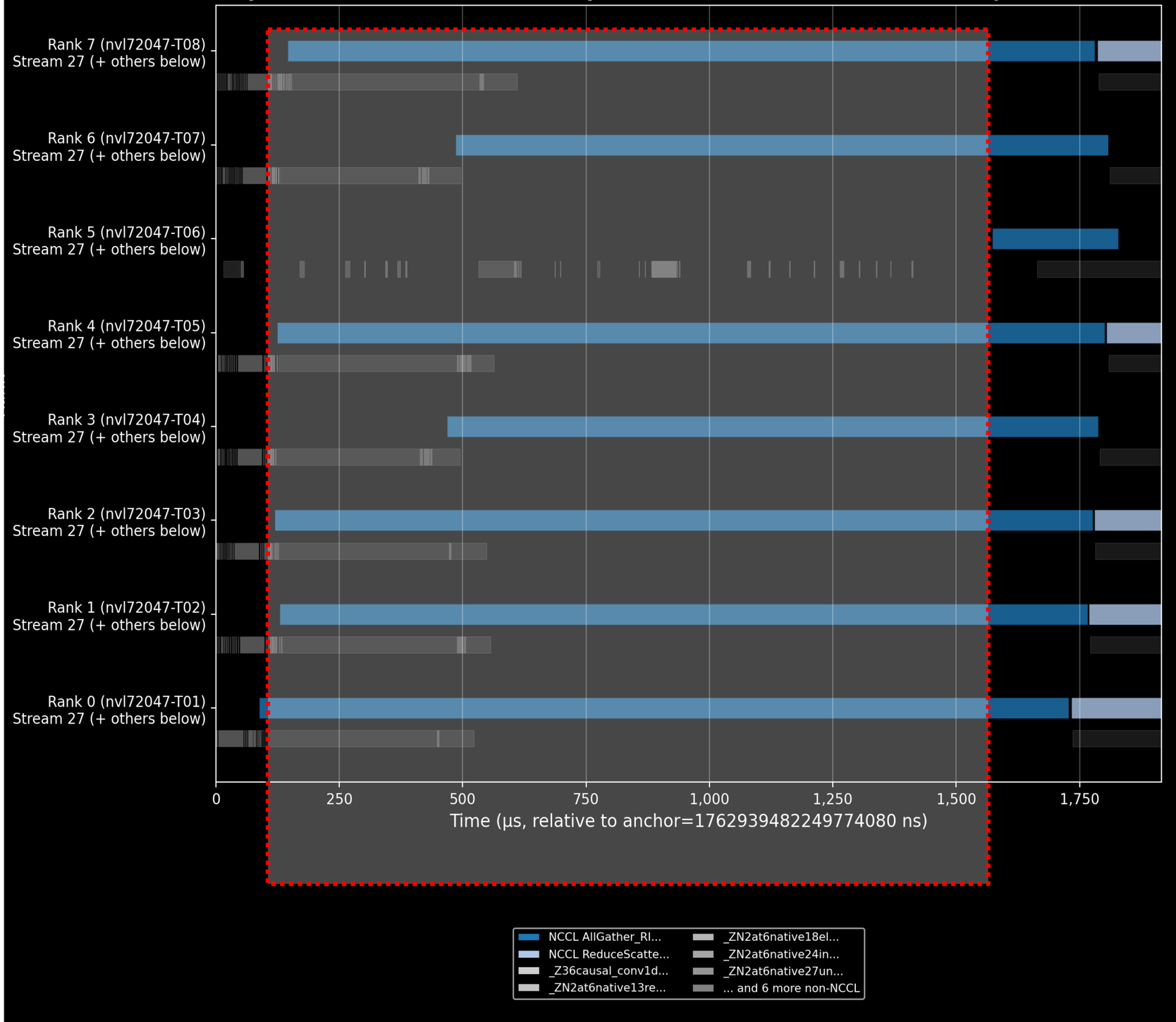





TIMESYNC
TO THE RESCUE!
WHOOSH!
NOW!
GPU METROPOLIS

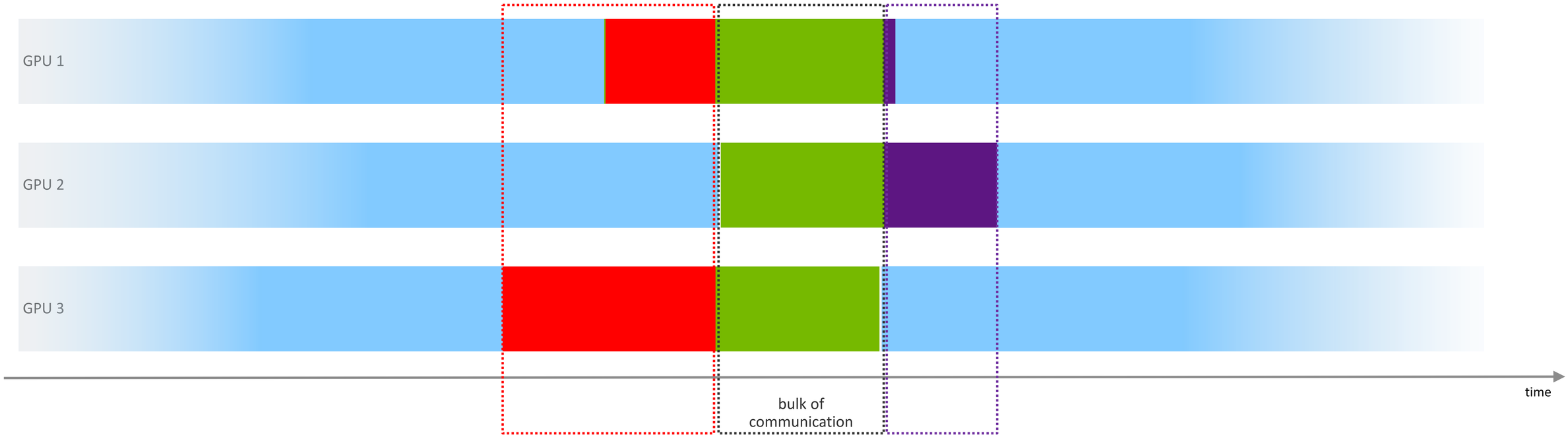
A small-scale example

- 47B parameter model pre-training.
- A simplistic configuration
- Rank #5 was artificially power-capped
- Even just ~10 usec time alignment made it painfully obvious that #5 was consistently late to join



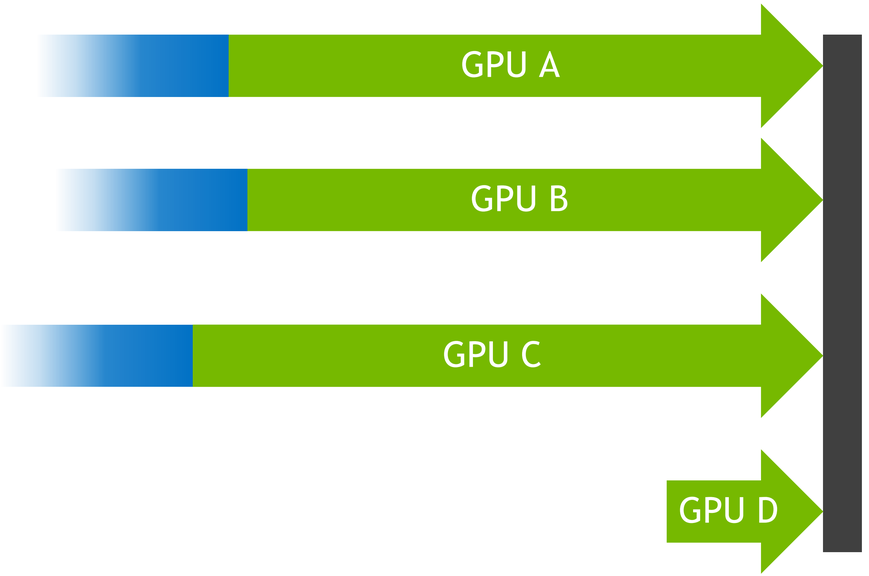
Timestamps, not duration

- **START** wait:
  - resources wasted collectively
  - a rank with long collective duration is **fast**
- **EXIT** delay:
  - resources wasted individually
  - a rank with long collective duration is **slow**



Without timesync, most analysis used start-wait

- Start-Wait's amplitude is order(s) of magnitude larger than Exit-Delay
- Many align the data by assuming that all GPUs exit simultaneously



BUT

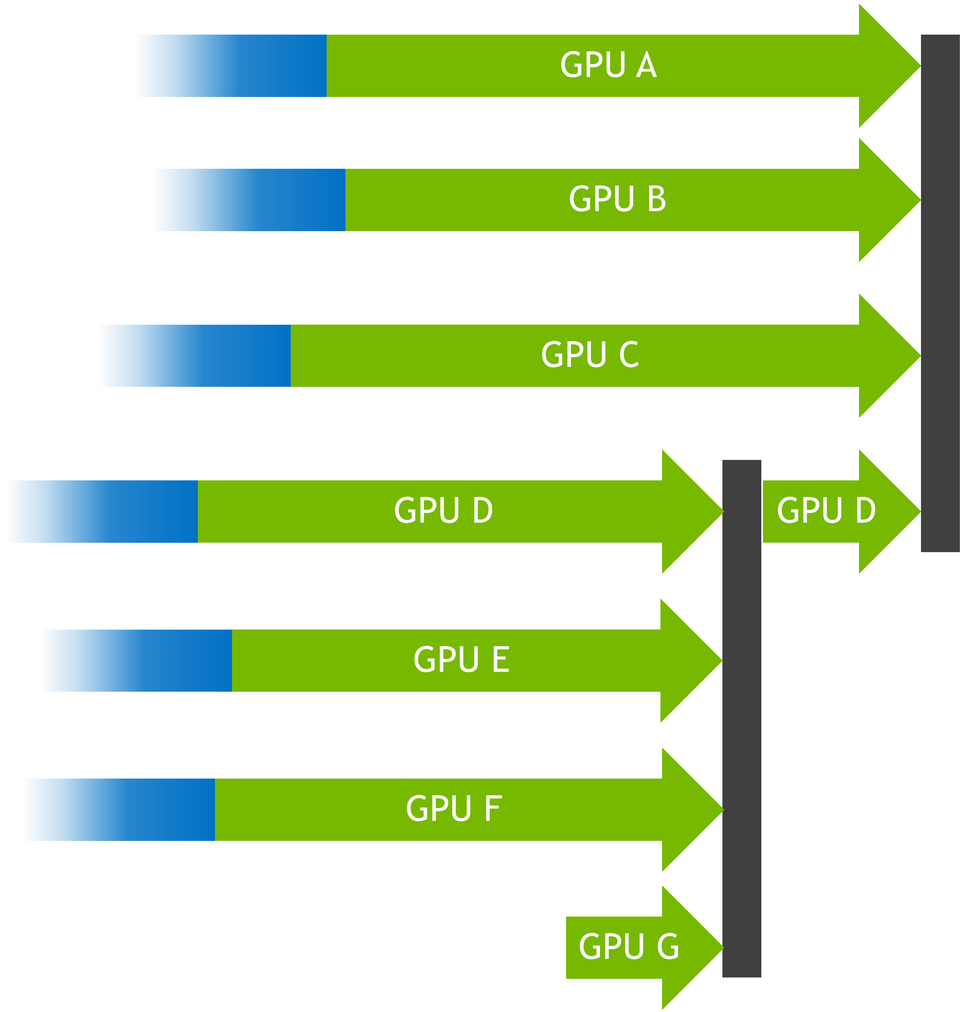
- They don't *really* exit simultaneously
- Just because a GPU is late to join the collective doesn't mean it's slow

Without timesync, most analysis used start-wait

- Start-Wait's amplitude is order(s) of magnitude larger than Exit-Delay
- Many align the data by assuming that all GPUs exit simultaneously

BUT

- They don't *really* exit simultaneously
- Just because a GPU is late to join the collective doesn't mean it's slow

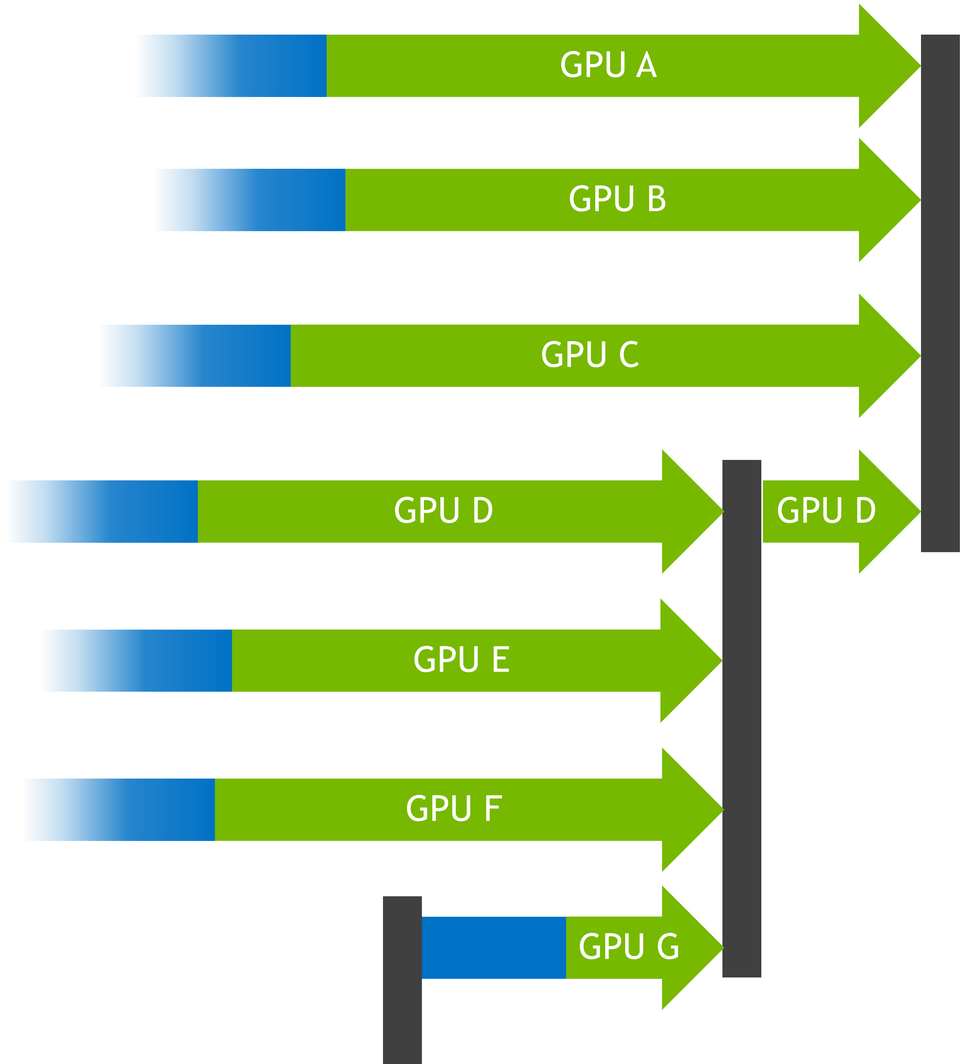


Without timesync, most analysis used start-wait

- Start-Wait's amplitude is order(s) of magnitude larger than Exit-Delay
- Many align the data by assuming that all GPUs exit simultaneously

BUT

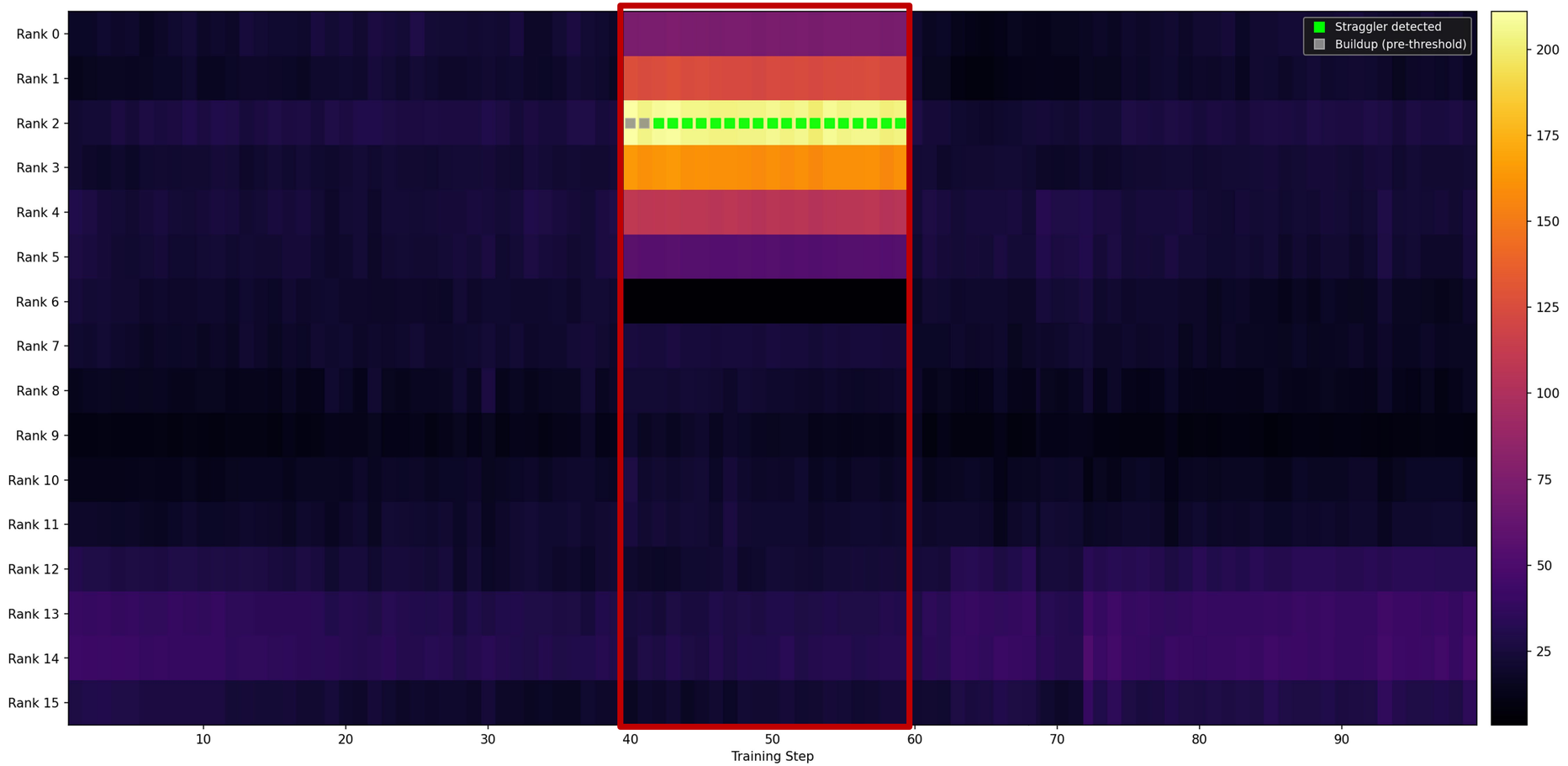
- They don't *really* exit simultaneously
- Just because a GPU is late to join the collective doesn't mean it's slow



Timesync enables new measurement

"If I'm slow, I'll be slow to exit as well"

- Per-collective, the differences are small - O(10 µs)
- You definitely need timesync to spot them



Heatmap of each rank's accumulated exit delay, per step

In AI datacenters, the rules are different

- Accuracy budget is ~microsecond
  - accuracy wrt. UTC/TAI far less critical than intra-cluster
  - holdover is seconds

- Compute-to-compute error is the goal
  - CPU <-> CPU
  - GPU <-> GPU

- The layout is different and dynamic
  - nodes cordoned off
  - jobs restarted
  - multiple parallel network links (bandwidth!)
  - switches with 100s-1000s of ports
  - even the network topology can change (optical switches)

What’s next?

Like Prometheus with fire, this community gives the world the foundational technology — and the world builds with it.

Today, we inspect. Tomorrow, we schedule.

Robust heuristics need robust timing data.

There's more to come.

Image generated by AI — fittingly.

Closing

If you build:

- sync silicon
- sync software
- PTP profiles
- timing standards
- verification equipment

AI datacenters are your next frontier, and the requirements are different. Come help shape them.

wwasko@nvidia.com ; gkorcia@nvidia.com

Find me at the break :)


NVIDIA