Novel Approaches to Network Telemetry

Essential for AI Performance

Networking

Novel Approaches to Network Telemetry: Essential for AI Performance

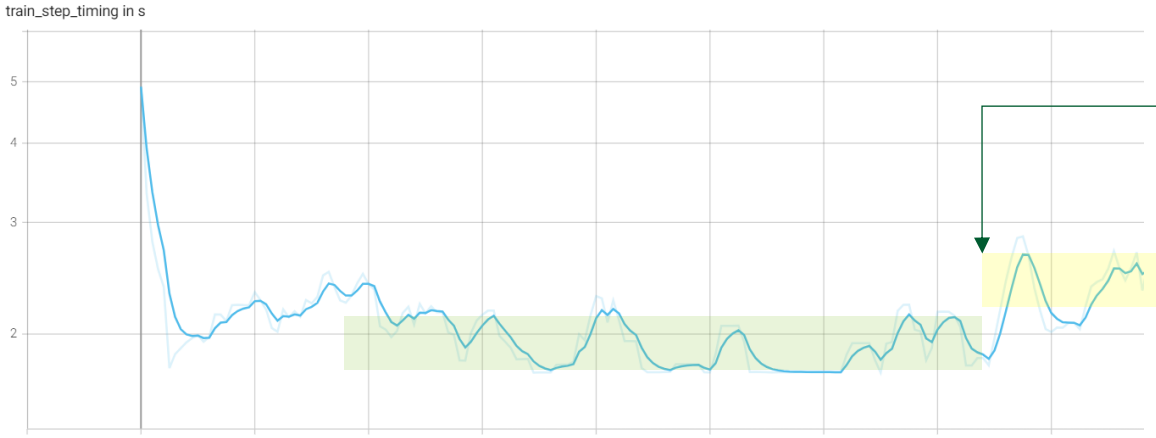
Roop Mukherjee, NVIDIA

“Just lost a full day!”

We are training an LLM with 128*8 GPUs.

Iteration time stabilizes around 1.75 seconds.

We expect to train for about 750K iterations, ~15 days of training.



After the first 250K iterations, ~5 days, the step time creeps up slightly- ~ 1.9s.

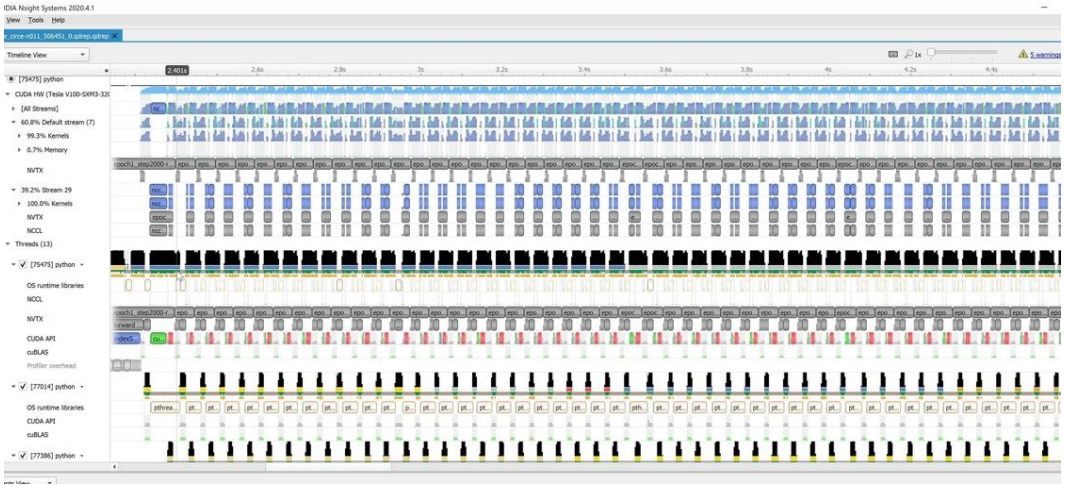
Our job completion time just got longer by 1 full day. Someone will ask: **Why?**

The Weakest Link of Large AI Networks

- .. It could be just 1 NIC out of 1024 that starts lagging the others slightly.
- AI training performance depends on tails; proceeds as quickly as its weakest link.
- GPT3 was trained on a very large cluster for that time. “10,000 GPUs and 400 gigabits per second of network connectivity for each GPU server”[1].
- Clusters as large as ~100K GPUs are being used for training everyday.
  - ~100K NICs, ~2.5K switches. Symmetrical topology, rail-optimized, adaptively routed around failures.
  - All ~ 16K paths between 2 GPUs look alike, so that they can be used elastically.
- Telemetry needs to find small divergences in any of those. Link down and server down are still necessary, but not sufficient. *Needle in the haystack!*

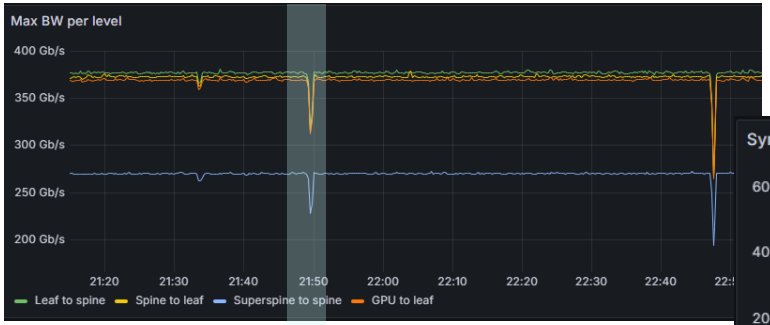
AI Traffic

- AI training traffic is vastly different from web requests.
  - Line rate TX/RX repeat like clockwork requiring very low latency.
  - Ranks in a job generate almost identical traffic.

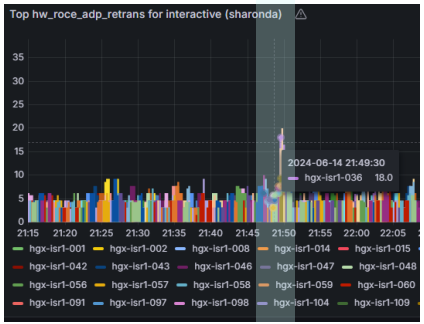


Current telemetry is designed to look at workloads and traffic with large statistical diversity. *When optimizing for AI training traffic, we look for similarity and synchronicity.*

Could We Have Saved the Day?

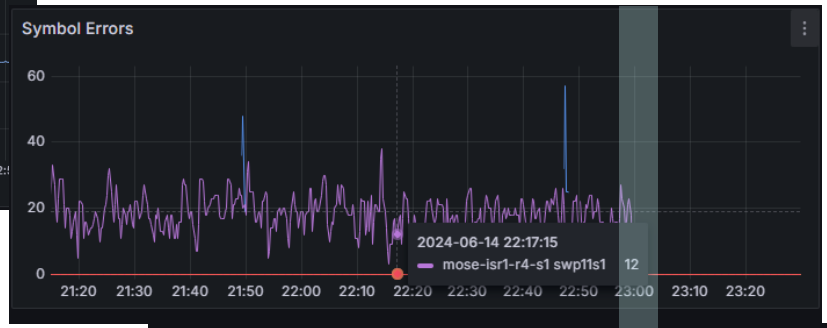


1. Drops in application performance seen.



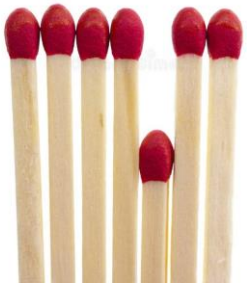
2. One host shows unusually high retransmissions.

3. Switch link to the host shows errors.



4. Disable link and return to stable performance.

Symmetry



- If all the hay should look similar, then picking the needle is easier.
- Our approach to telemetry in large AI clusters is to exploit symmetries.
  - Symmetry in fabric means that instead of looking for each leaf link, we use the expectation that *each link, when functioning well, should have almost the same data rate*. p01 ≈ p50 ≈ p99. When some link does not, it's worth checking on.

In the happy case, p01 ~ max:

- Workload sends equal traffic to all rails.
- AR balances load well.

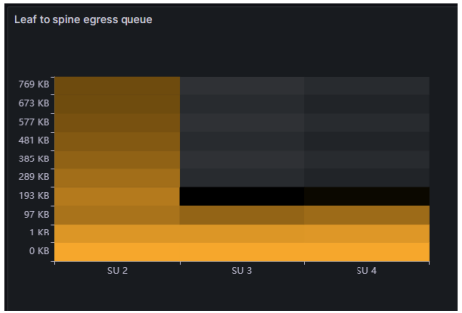
Credit: Thomasiecker / Dreamstime

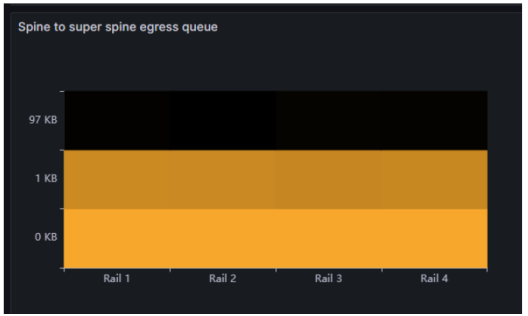
Symmetry Allows Aggregation

- Instead of plotting uplink queues of all leaf switches we can plot a bar for the whole SU
- Instead of plotting all queues of all spines towards the super spines, we plot one per rail
- Alerts let us know if there are outliers and not to use a particular aggregation.

For each bar, since traffic repeats, we compress in time dimension by using histograms.

Plot buckets side-by-side, shade indicates frequency (like heatmaps).

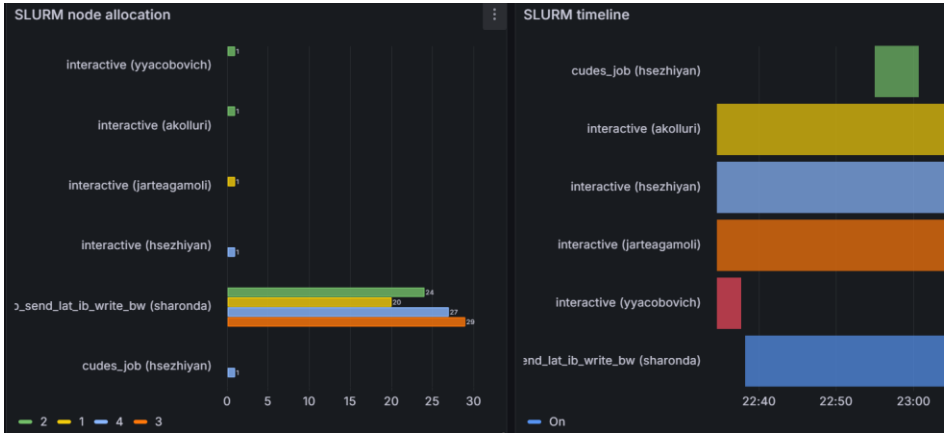




- Horizontal bands emerge, indicating symmetry.
- Breaks in the bands indicate deviations.

Symmetry in Jobs

- Hosts participating in a job have almost identical data Tx/Rx
- **Filtering** for job participants removes unrelated dissimilarities
- **Allocation view** of jobs shows proportion traffic crossing spine, super-spine levels, where they compete with other jobs
- **Timeline view** of jobs indicates which jobs' traffic may interact



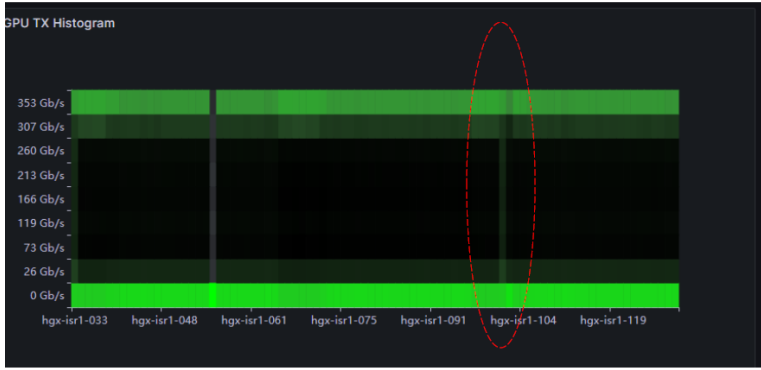
Symmetry Makes the Needle Stand Out

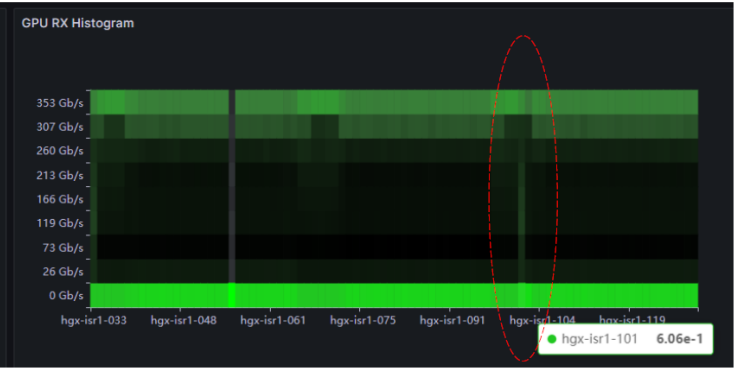
AI training traffic symmetry:

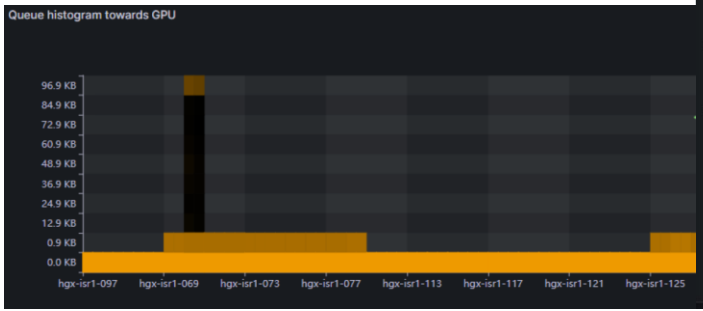
Tx/Rx at either at full rate or 0: bimodal

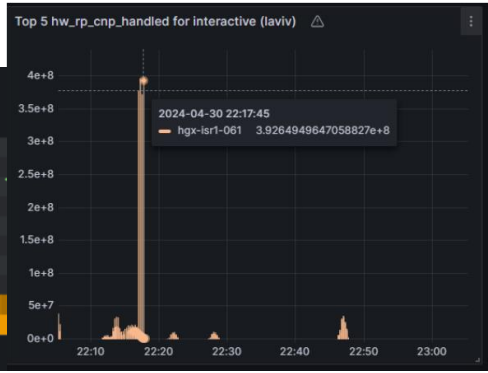
Queue buildup, CNPs and timeouts should be similar for ranks in a job

The outliers stand out- point us where to debug or tune









Making AI Networks Tractable

- Traditional telemetry is not suitable for AI network to
  - Debug issues over such scales, nor improve efficiency of distributed computing AI workloads
- AI networks require workload-specific debuggability and tunability
- We presented our experience
  - How to understand telemetry at scale
  - How to debug and tune large AI workloads
- We are incorporating these tools into the Spectrum-X stack.
  - Spectrum-X Platform - https://www.nvidia.com/en-us/networking/spectrumx/
  - Spectrum-X Video - https://www.youtube.com/watch?v=nKqfi3q4S5I
  - BlueField DPU - https://www.nvidia.com/en-us/networking/products/data-processing-unit/
  - Spectrum Switches - https://www.nvidia.com/en-us/networking/ethernet-switching/
  - NVIDIA SAI - https://developer.nvidia.com/networking/ethernet-switch-sdk

Thank you!