Observation and Control of Distributed AI/ML Workloads in the Public Cloud using Clockwork

Vinay Sriram
Senior Software Engineer

The AI Conference 2023

Clockwork Systems
Sept 26th, 2023

CLOCKWORK.IO

Talk Overview

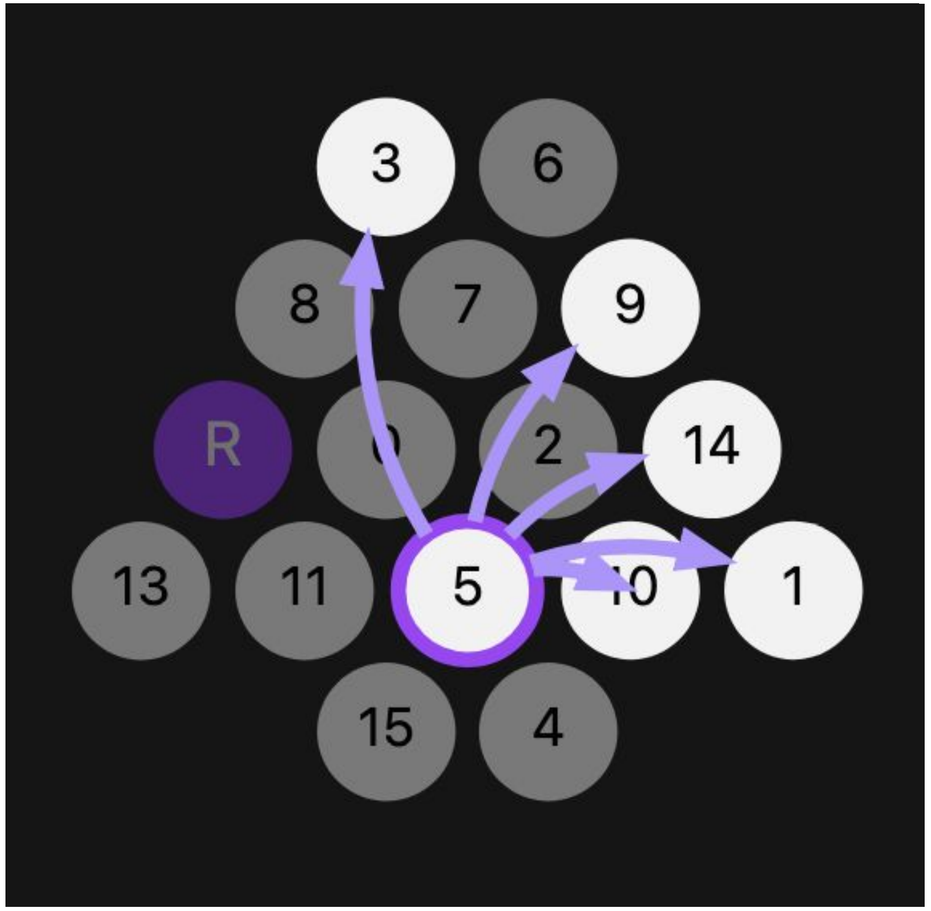
- **Clockwork's Product Suite:**
  Tools for edge-based observation and control

- **Observing Distributed AI/ML Workloads:**
  Analyzing NVIDIA's model training communications primitives using Clockwork's technology

- **Speeding Up Distributed AI/ML Workloads:**
  Clockwork's 2-step solution and evaluation on common benchmarks

Clockwork’s Product Suite

Clockwork's Product Suite



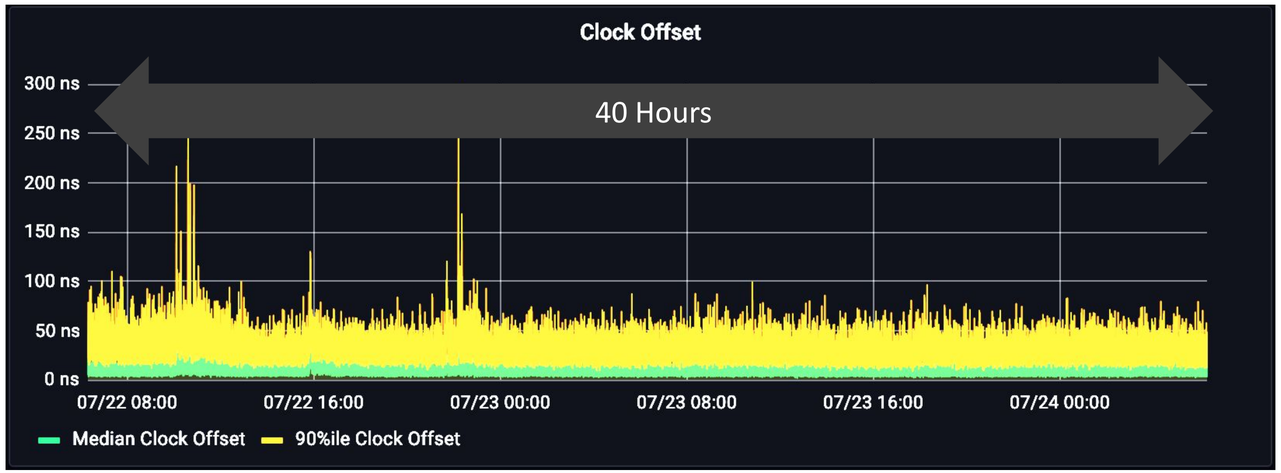
- The CW agent is installed on each machine.
- **Monitor:** *Latency Sensei* sets up a low-overhead, always-on "probe mesh" connecting each VM to 5 others.
  - Sync *all VM clocks highly accurately* to a common reference [1]
  - Measure one-way-delays precisely
  - Detect VM colocation
- **Control:** Based on accurate one-way-delay measurements, *Packet Rocket* can sense and control TCP traffic, providing a "zero-drop" network [2]
- **No hardware support or upgrades needed!**

[1] Clock Sync: https://www.usenix.org/system/files/conference/nsdi18/nsdi18-geng.pdf

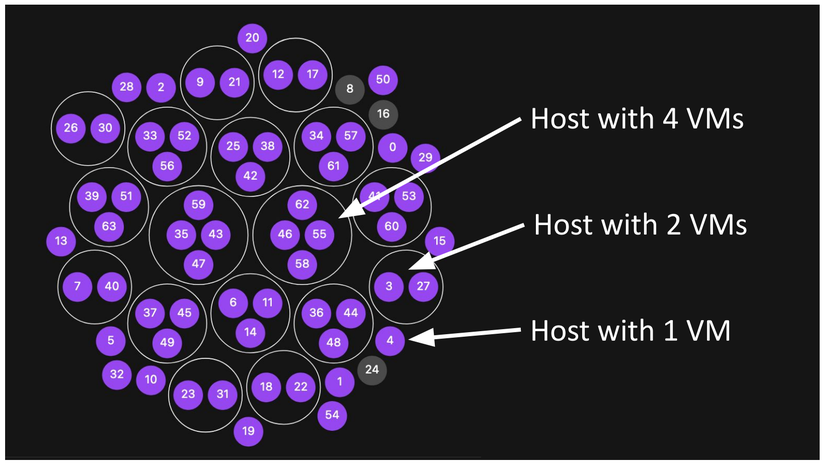
[2] Control: https://www.usenix.org/system/files/nsdi21-liu.pdf

Clockwork's Product Suite

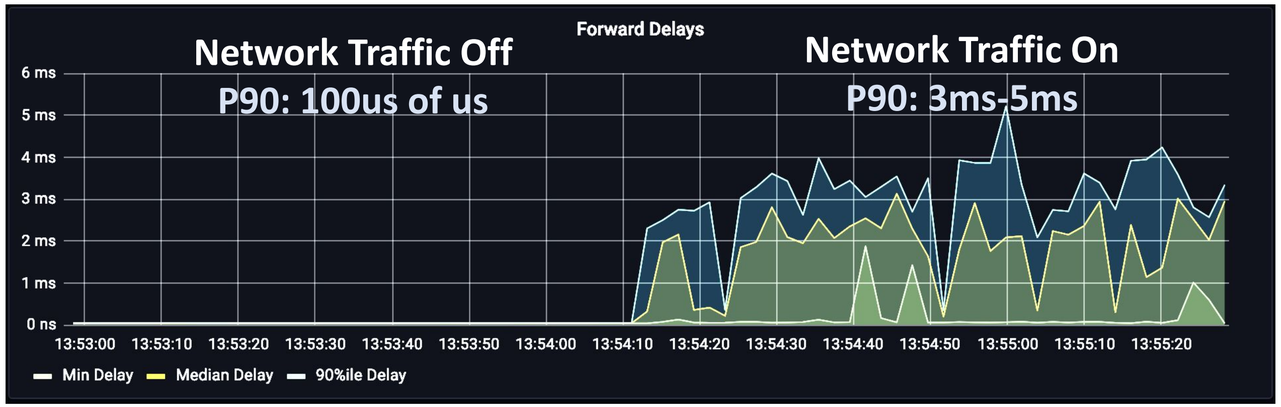
Accurate Clock Synchronization (100s of ns)



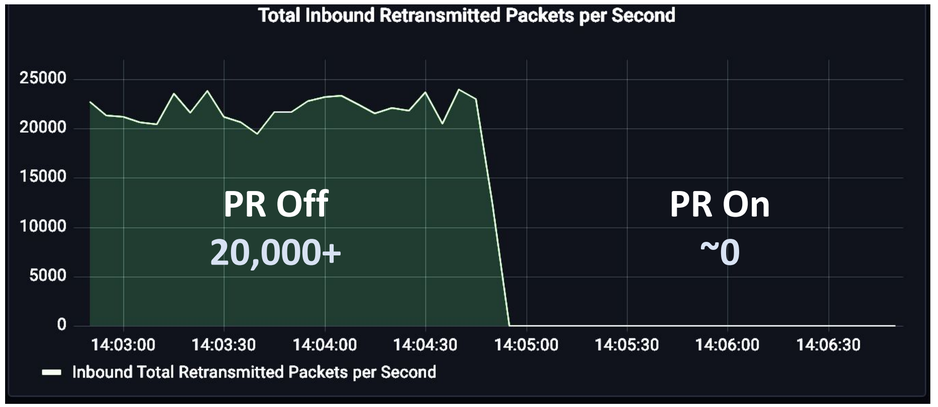
VM Colocation Detection



Precise One-Way-Delay Measurements

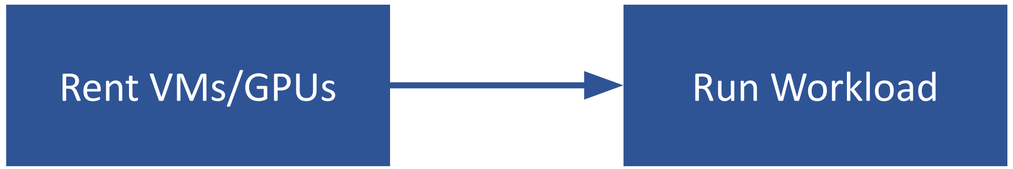


Packet Rocket: "Zero-Drop" Network

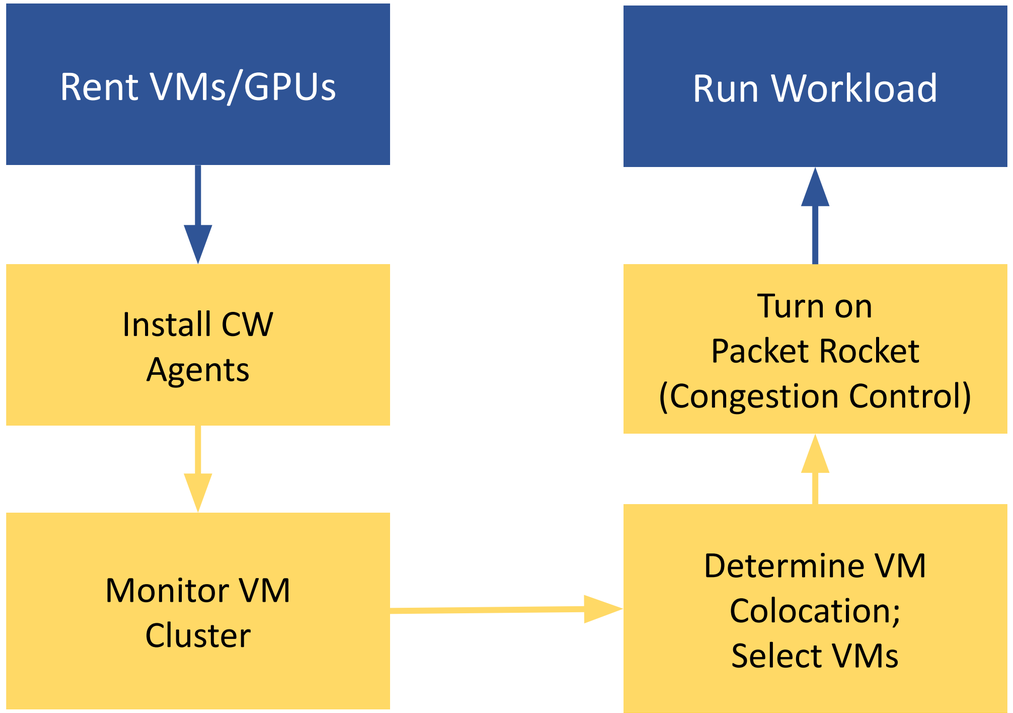


Using Clockwork for Distributed Training

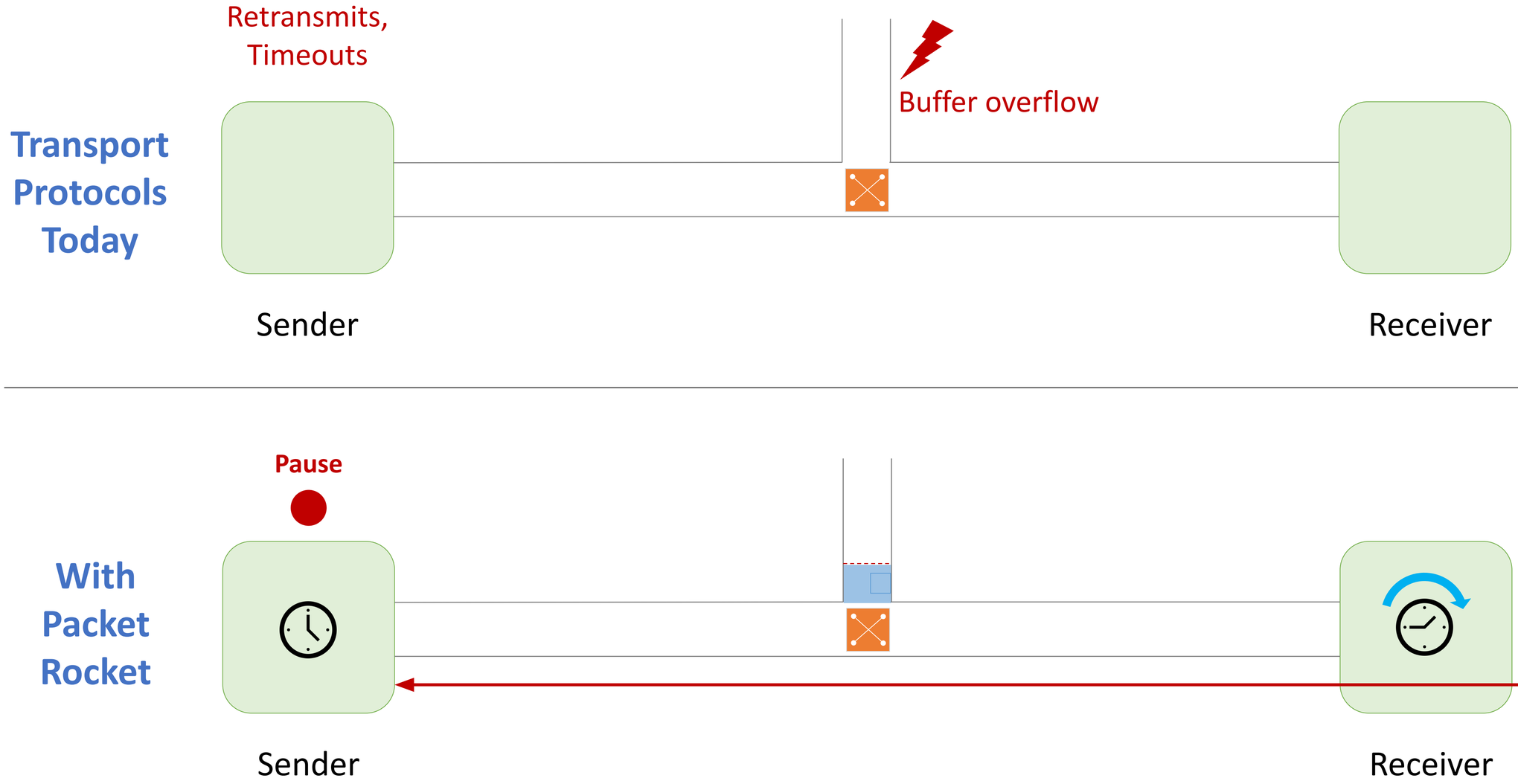
Current Practice for Cloud Tenants:



With Clockwork's Software:



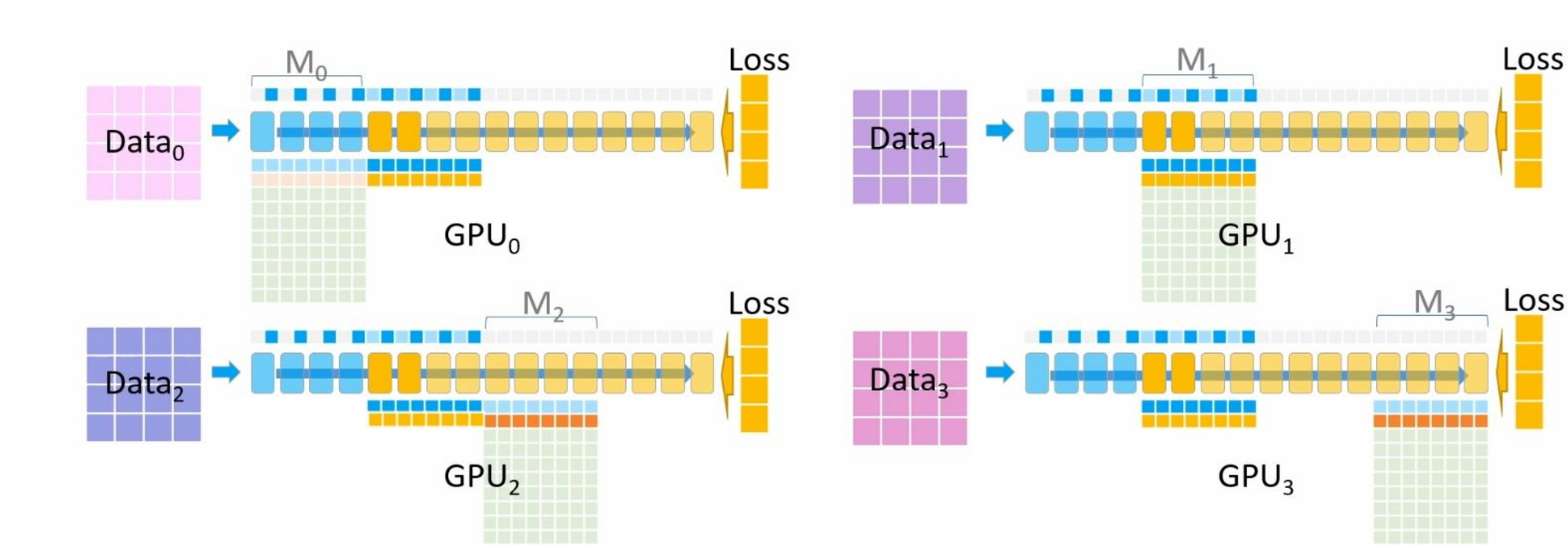
Packet Rocket: Congestion Control at the Edge



Observing AI/ML Workloads Distributed Across Hosts

NVIDA NCCL Primitives

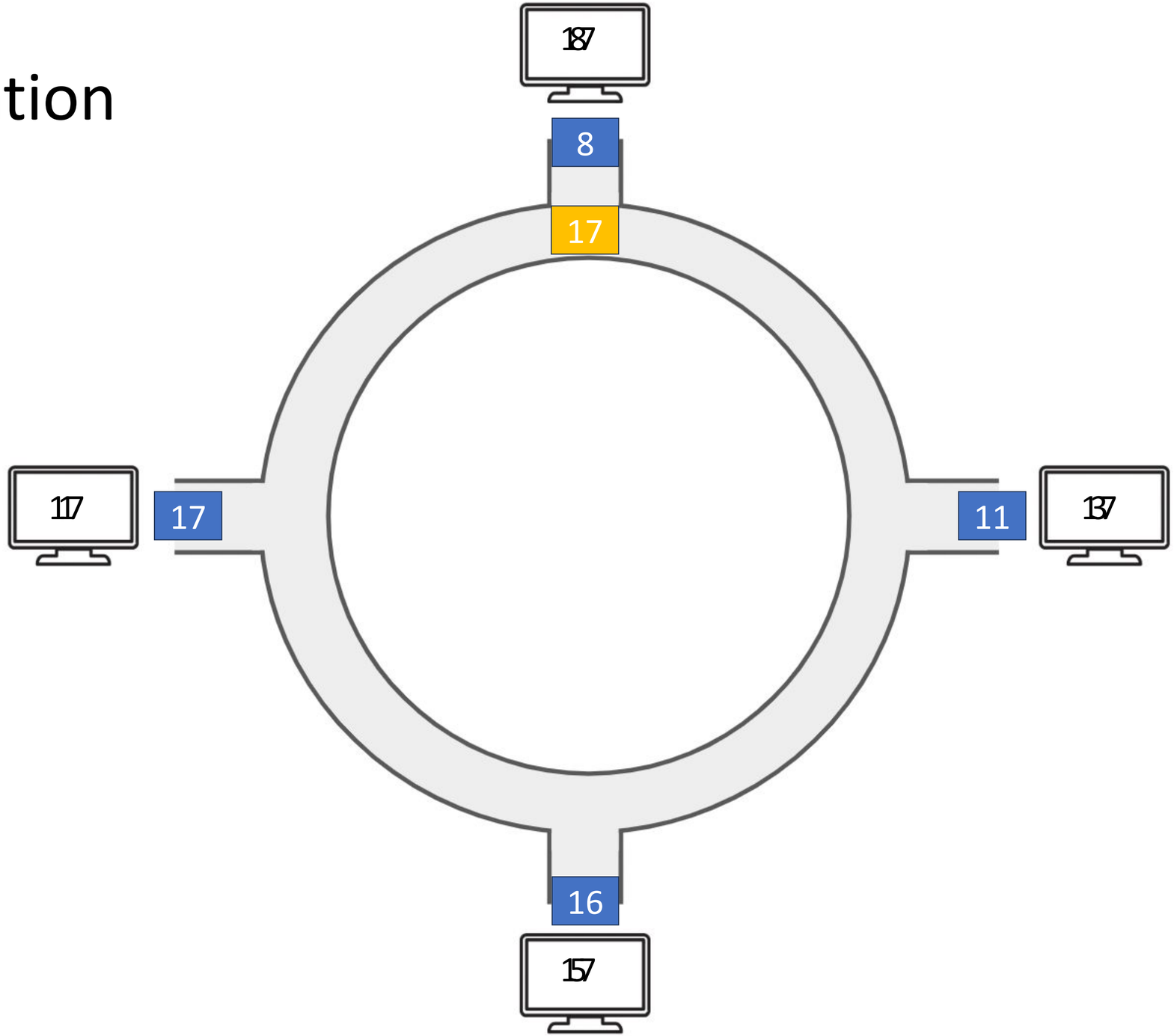
- NVIDIA's NCCL implements application **primitives** relevant for large model training (used, for example, in Microsoft's DeepSpeed [1]). These include all-reduce, all-gather, reduce-scatter, etc.



[1] **https://www.microsoft.com/en-us/research/blog/zero-deepspeed-new-system-optimizations-enable-training-models-with-over-100-billion-parameters/**

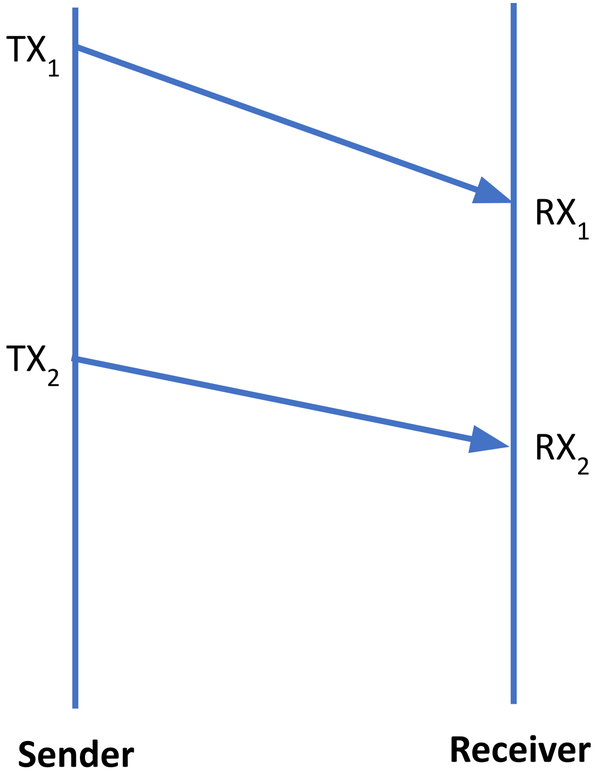
NCCL Ring Implementation

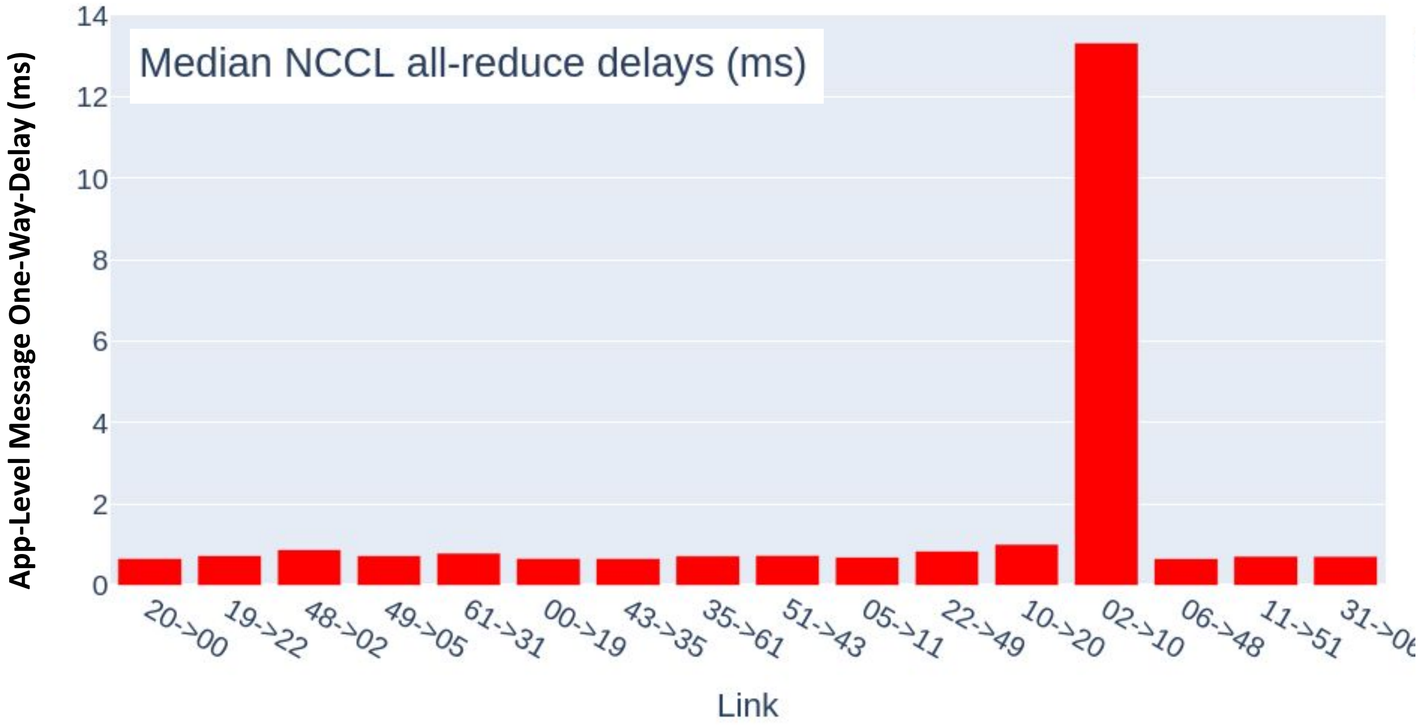
- NCCL primitives are often executed using a ring-style communication pattern
- In **all-reduce**, components of the sum get passed around the ring until the sum is computed. Then, everyone gets the sum.
- The collective operation finish time is determined by the sum of the ring latencies



Accurate time enables accurate app-level measurements.

- We have instrumented **NCCL** to log individual message send and receive timestamps.
- Accurate clocks allow us to **match these into one-way-delay measurements.**

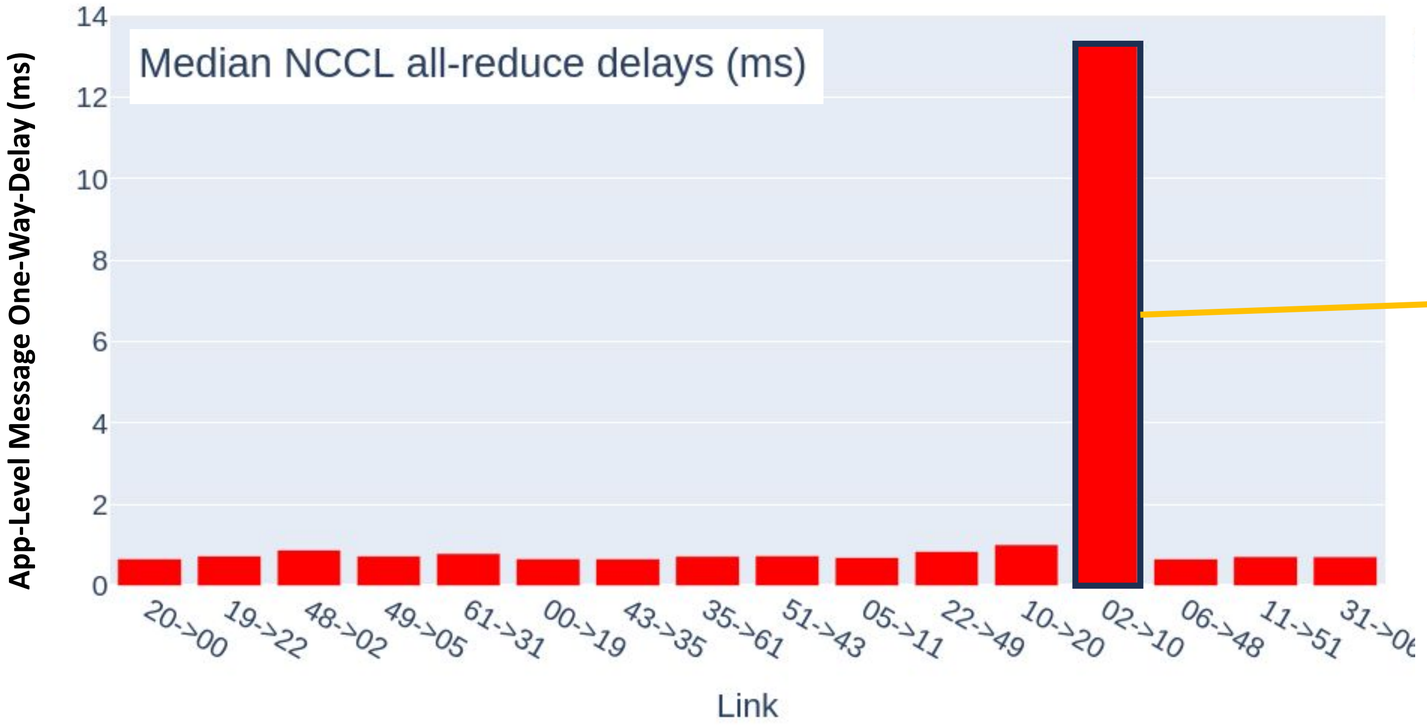


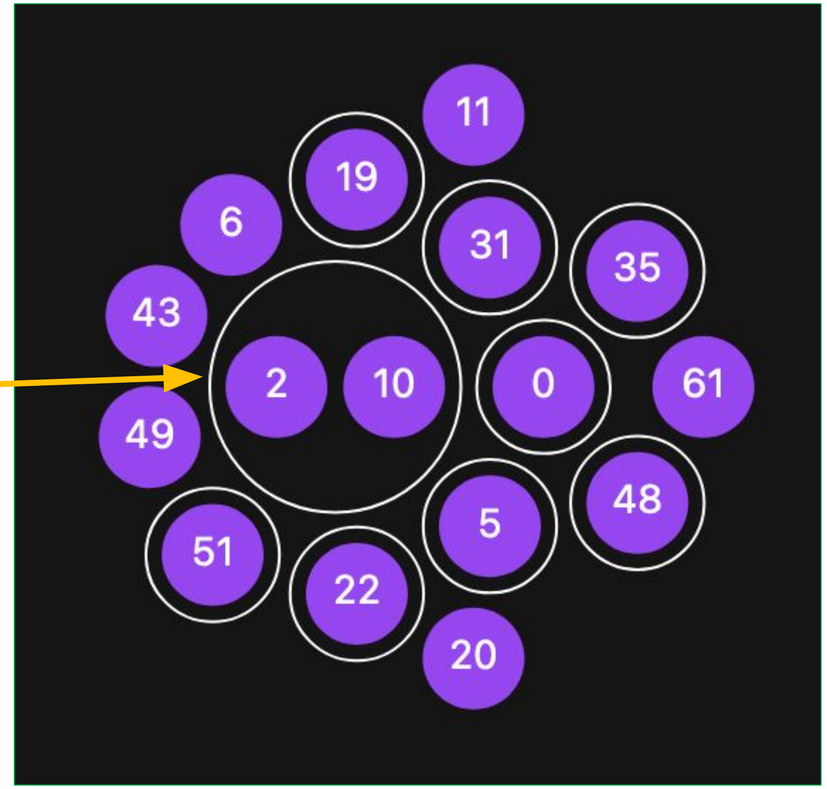


Measurements on one-way delays can identify bottleneck links.

Clockwork’s colocation detector can explain bottlenecks.

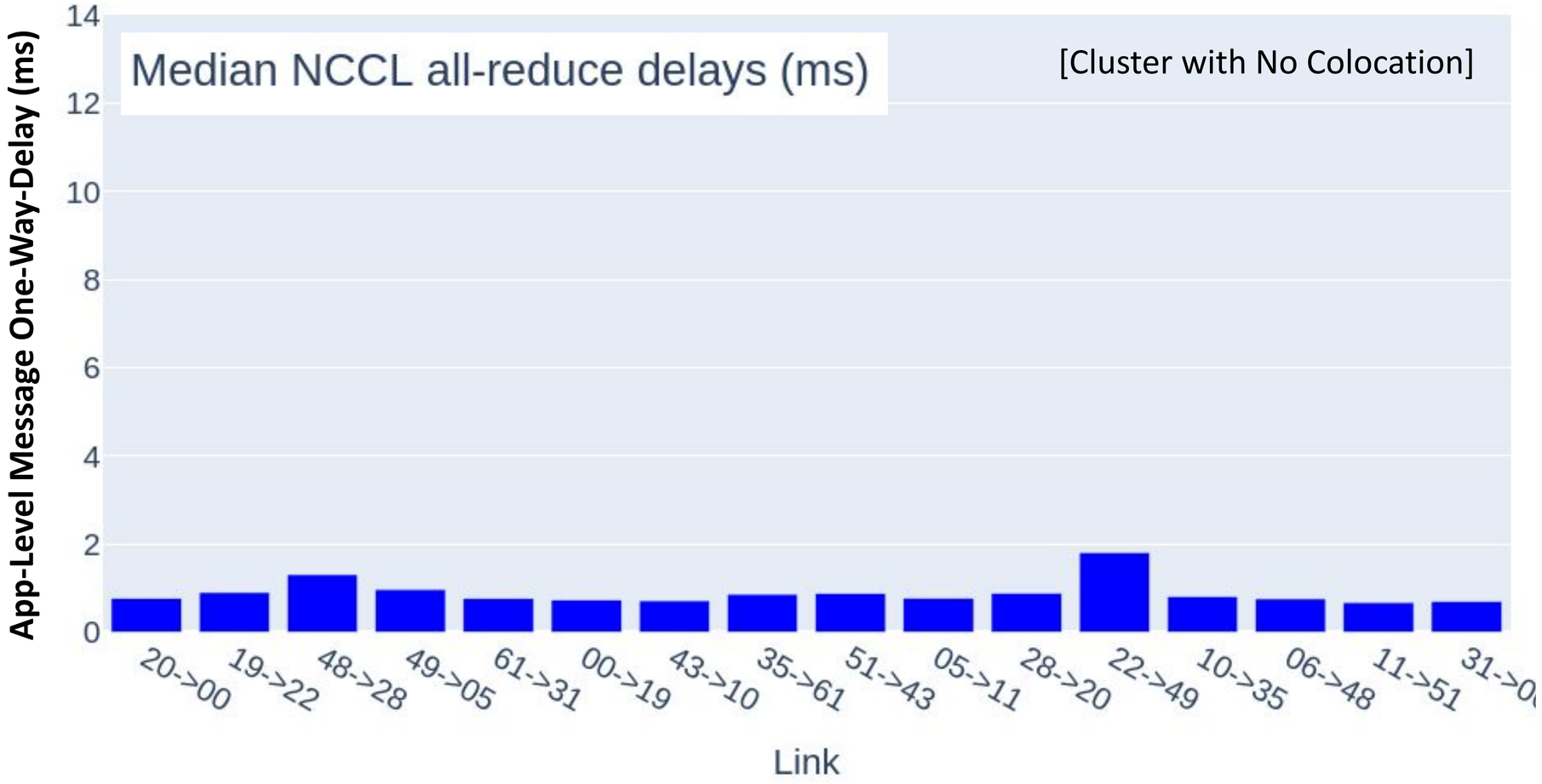
- Colocated VMs can contend for hypervisor/physical machine resources, creating high queuing delays on the sender.





Clockwork’s colocation detector can explain bottlenecks.

- In the absence of colocation, the throughput bottleneck causing high queueing delay does not exist.



Speeding Up AI/ML Workloads Distributed Across Hosts

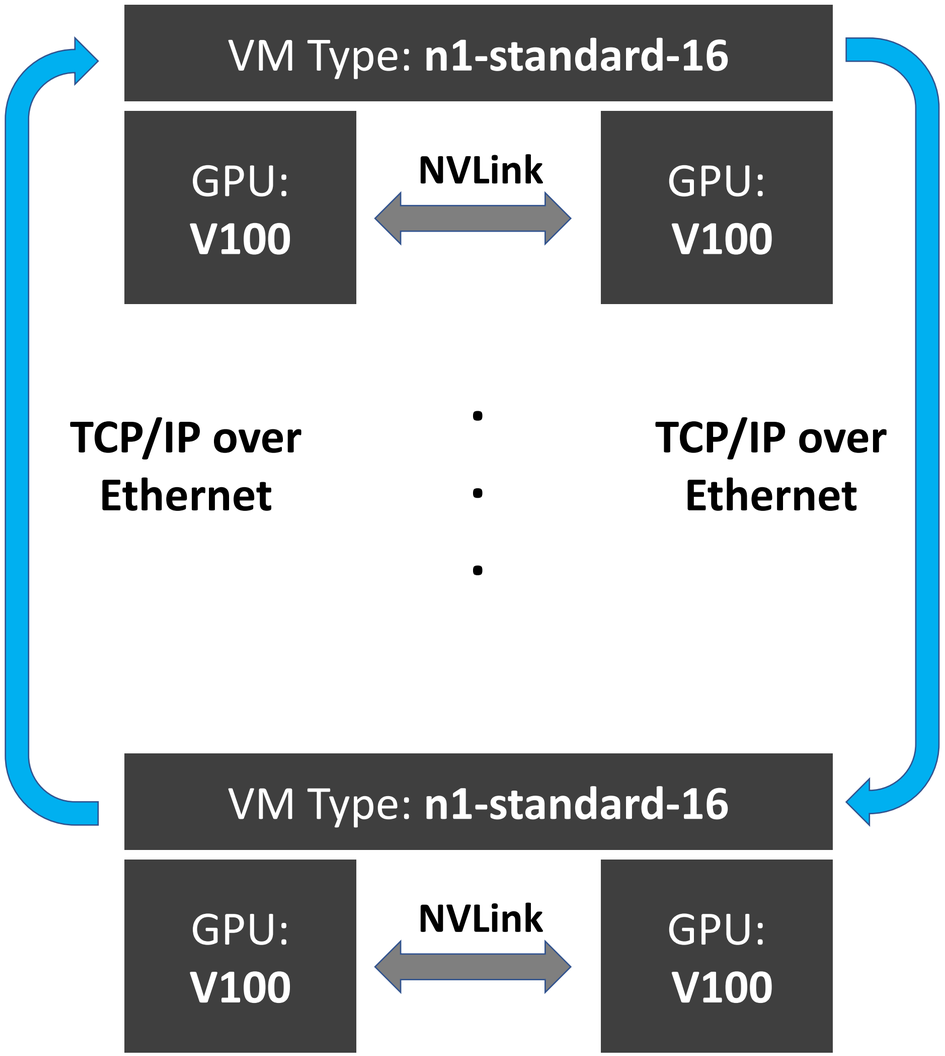
Evaluation on NCCL Benchmarks: Google Cloud

Experimental Setup

We ran nccl-tests (https://github.com/NVIDIA/nccl-tests), a popular set of benchmarks released by NVIDIA, on Google Cloud Platform.

Traffic between GPUs attached to the same host uses NVLink, traffic between GPUs attached to different hosts uses TCP/IP.

The experiment clusters contained 16 Hosts, each attached to two V100 GPUs.



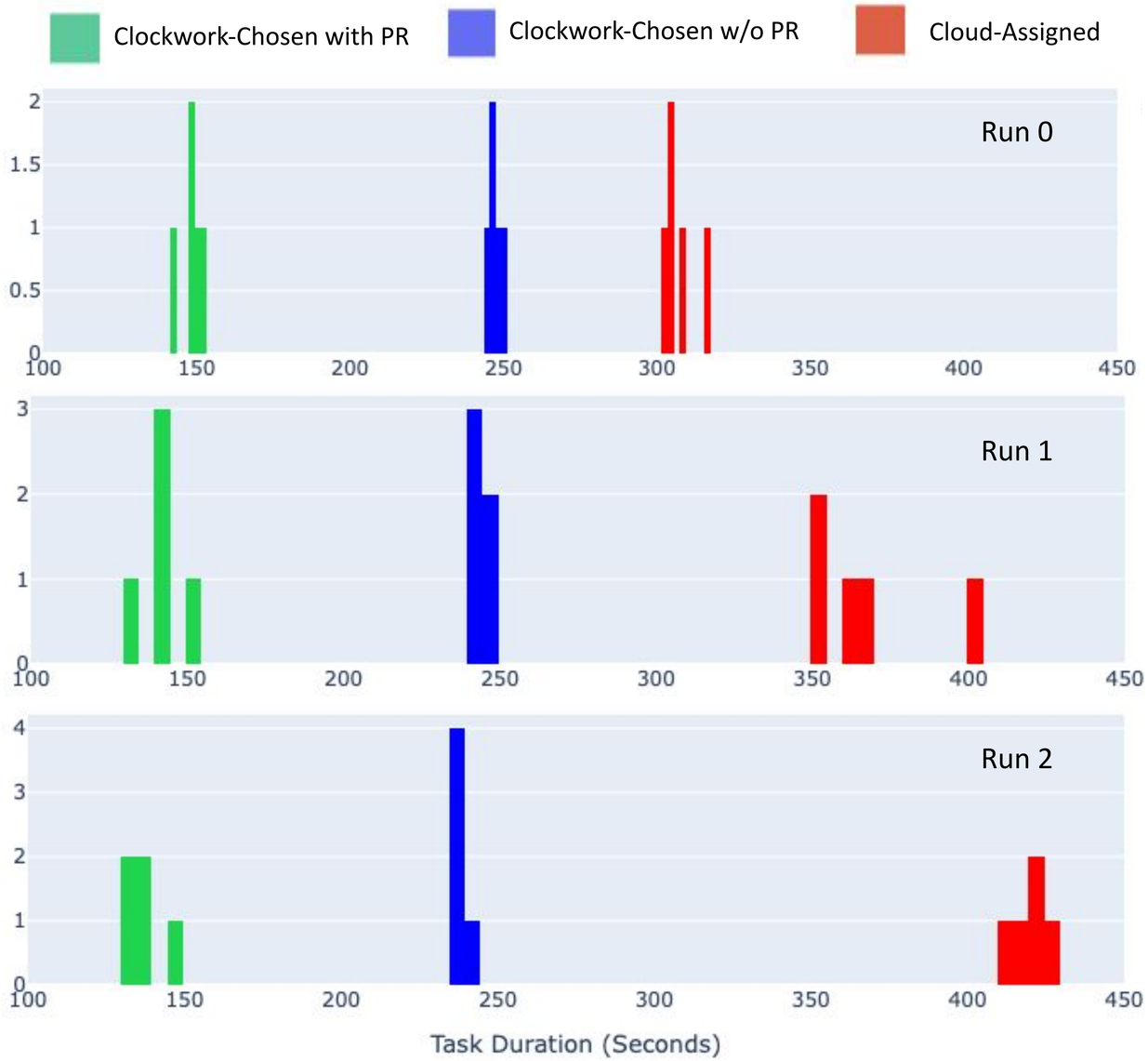
Performance on the **all-reduce** Communication Primitive

Created 3 clusters of each type and ran the all-reduce task 5 times on each cluster.

Compared to a cloud-assigned cluster:

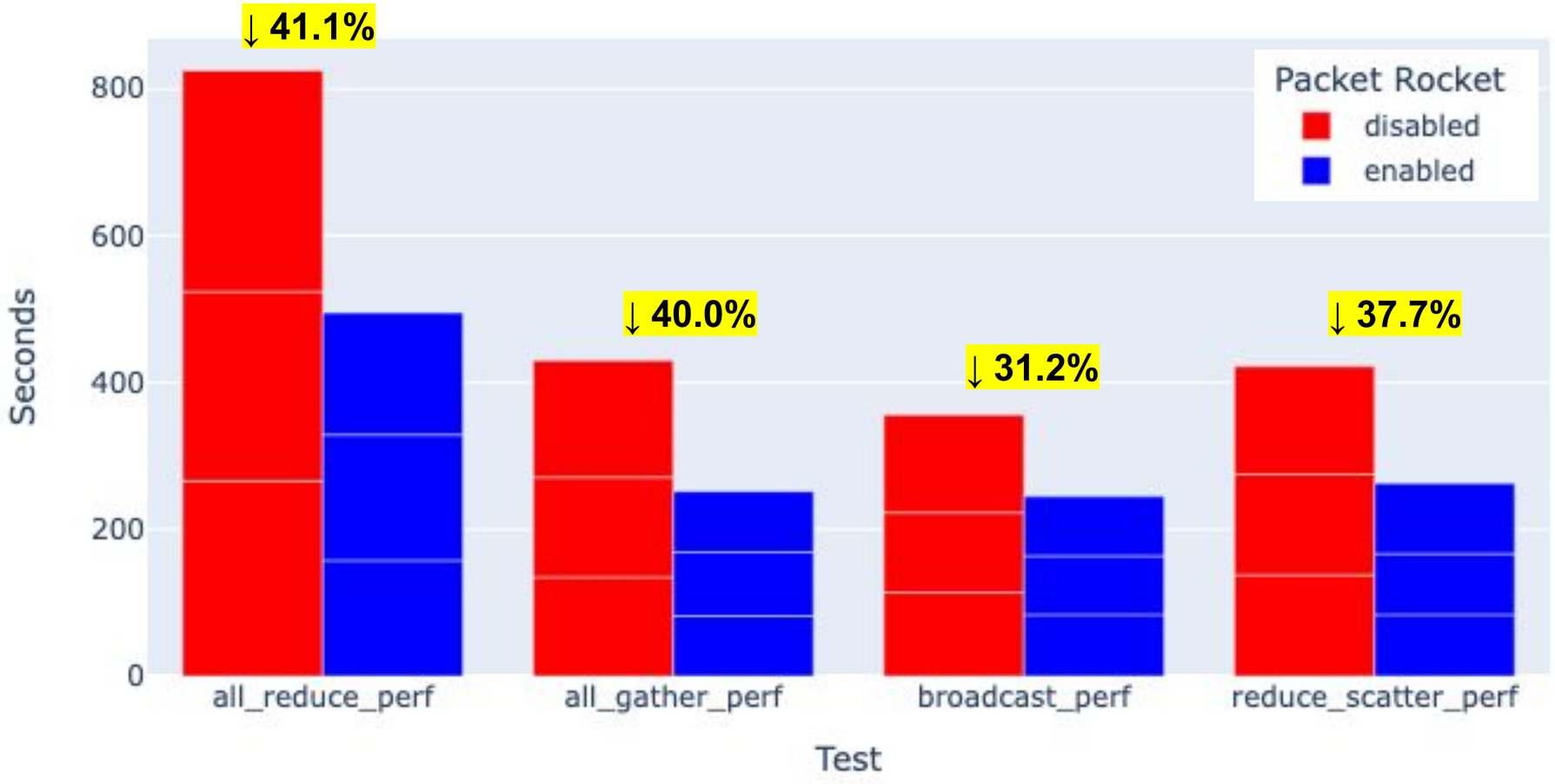
- A Clockwork-chosen cluster **decreases by 33%** the median task completion time.
- A Clockwork-chosen cluster with Packet Rocket **decreases by 61%** the median task completion time; equivalently, TCP throughput **increases by 2.6x**.

Clockwork's 2-step solution also **reduces the variance** in run times dramatically.



Packet Rocket improves completion time **across the board** on GCP

[CW-Chosen Cluster] Packet Rocket alone produces a 31-41% decrease in benchmark completion time.



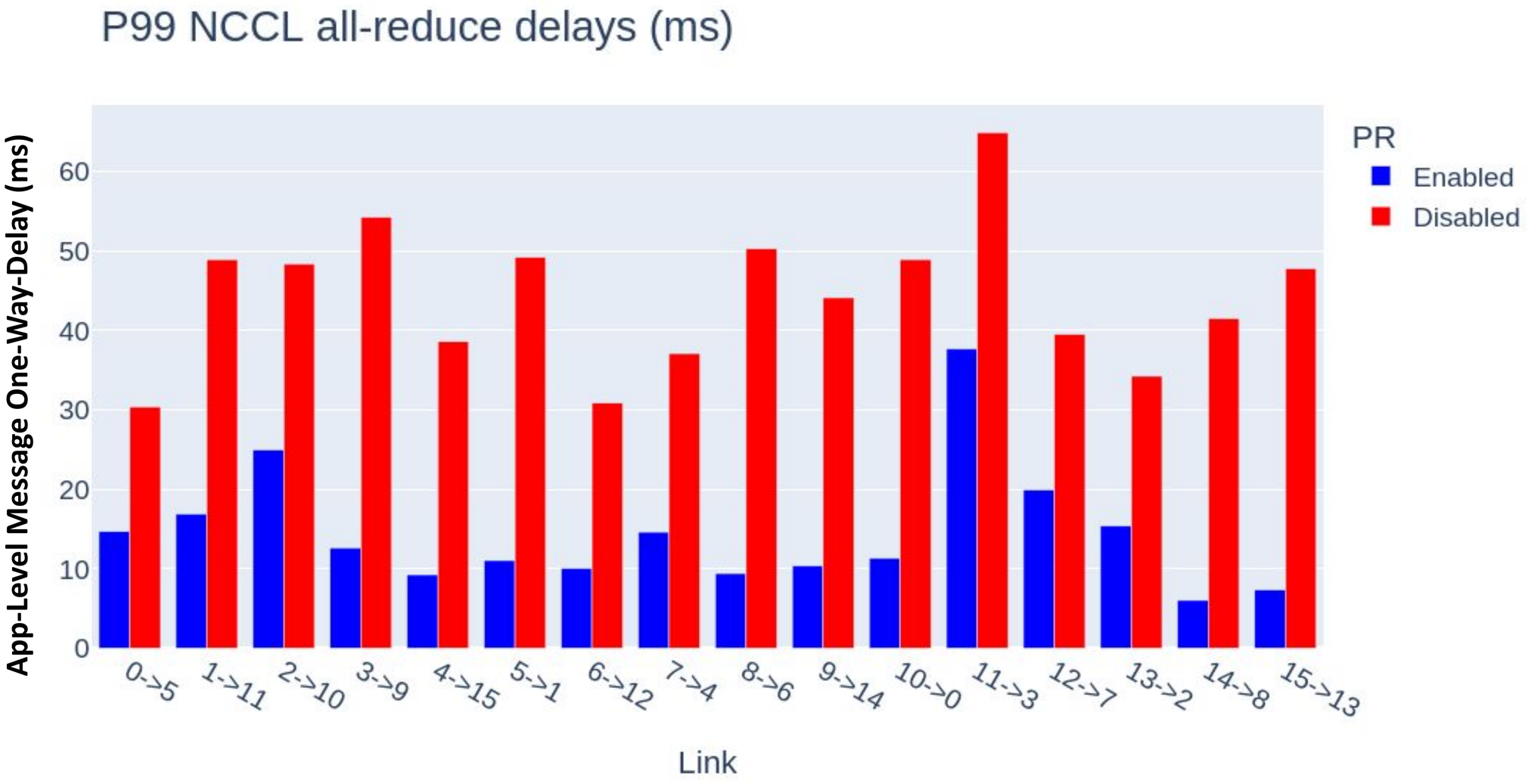
On each of the four NCCL tasks: (1) We ran 3 trials with PR and 3 trials without PR for 100 iterations of each.
(2) There were two NCCL connections per node (one per GPU), and each NCCL connection used 10 threads.

Performance on the **all-reduce** Communication Primitive

[CW-Chosen Cluster] Packet Rocket **eliminates drops** and **significantly reduces delays**.

Packet Rocket Off | **Packet Rocket On**

Packet Rocket reduces app-level delays.



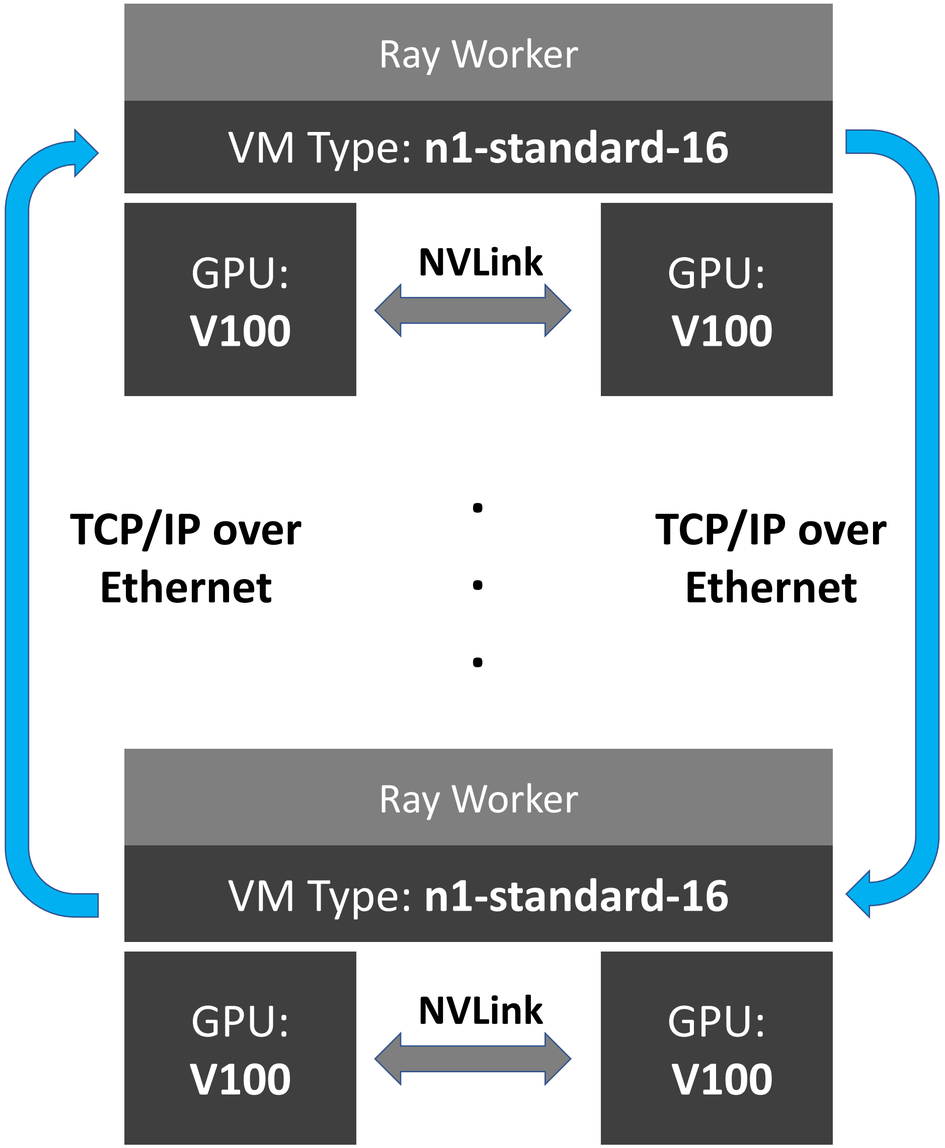
Evaluation on the Ray Collective Communications Library: Google Cloud

Experimental Setup (Ray on GCP)

We ran the Ray collective communications library (https://docs.ray.io/en/latest/ray-more-libs/ray-collective.html), which provides a set of primitives for Ray. NCCL is used as the communications backend for GPUs.

Traffic between GPUs attached to the same host uses NVLink, traffic between GPUs attached to different hosts uses TCP/IP.

The experiment clusters contained 16 Hosts (one head node and 15 additional workers), each attached to two V100 GPUs.



Performance Benefit on Ray CCL on GCP

Created 2 clusters of each type and ran the all-reduce task (for 28 iterations) 3 times on each cluster.

Compared to a cloud-assigned cluster:

- A Clockwork-chosen cluster **decreases by 7.9%** the median task completion time.
- A Clockwork-chosen cluster with Packet Rocket **decreases by 26%** the median task completion time.

Clockwork's 2-step solution also **reduces the variance** in run times.

Key Takeaways

- Performance Optimizations for Distributed ML Workloads is a Hot Area:
  - Hardware acceleration on GPUs and other accelerators
  - Development of fundamental communication libraries (e.g., NCCL)
  - Development of compute frameworks such as Ray

- Clockwork's product suite can provide insights into network bottlenecks and increase observability

- Preliminary results from tests on Google Cloud have shown that Clockwork's 2-step solution can significantly accelerate NCCL communications benchmarks (by 30%-60%)

- When training large models on GPUs across hosts, communication can become a bottleneck in TCP/IP + Ethernet networks. **Clockwork's tech can help here.**

- **Looking for beta users for validation and collaboration!**