ASPLOS 2026

Fine-grained and Non-intrusive LLM Training Monitoring via Microsecond-level Traffic Measurement

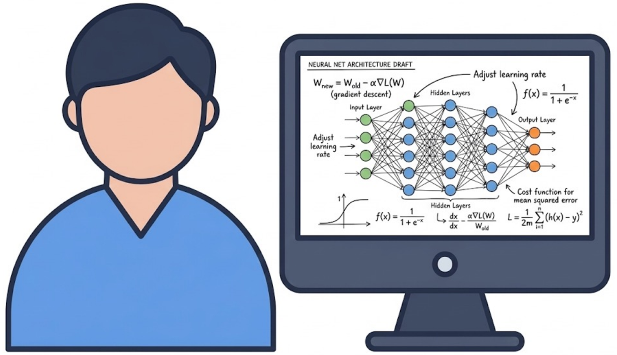
Yibo Xiao, Hao Zheng, Haifeng Sun, Qingkai Meng, **Jiong Duan**, Xiaohe Hu, Rong Gu, Guihai Chen, Chen Tian

南京大學
NANJING UNIVERSITY

NUS
National University of Singapore

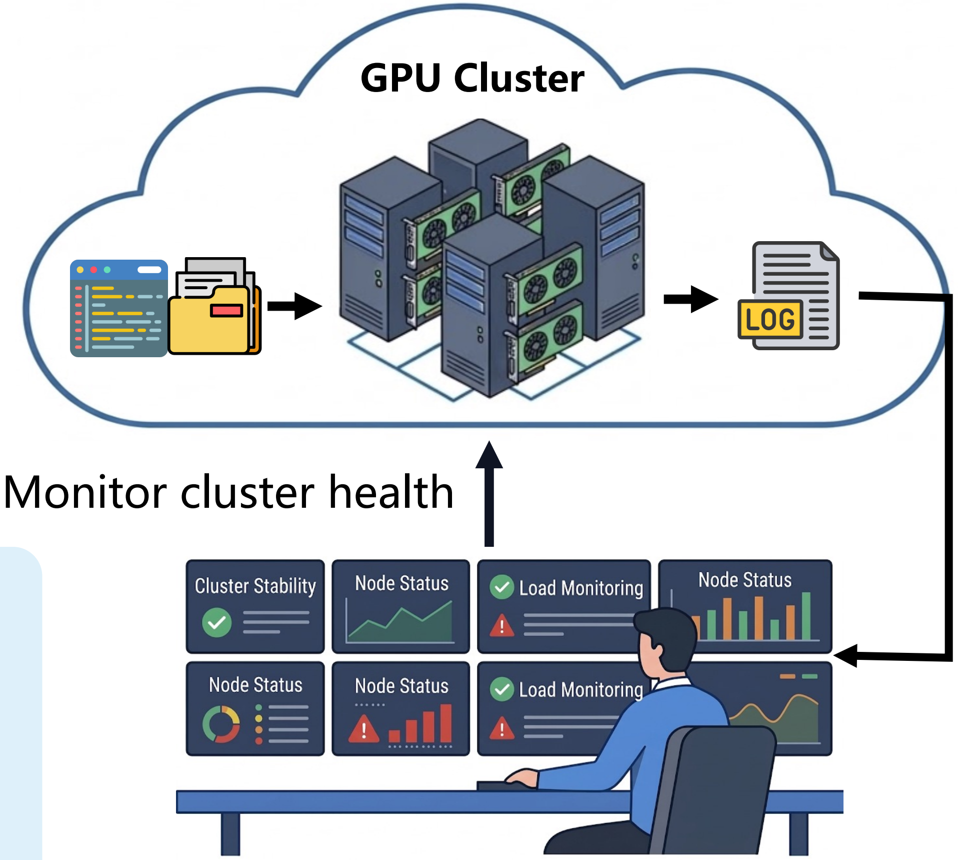
基流科技
Infrawaves

LLM Training as a Cloud service



Model Developer

Submit training job



Cloud provider

Public AI Training Service Paradigm

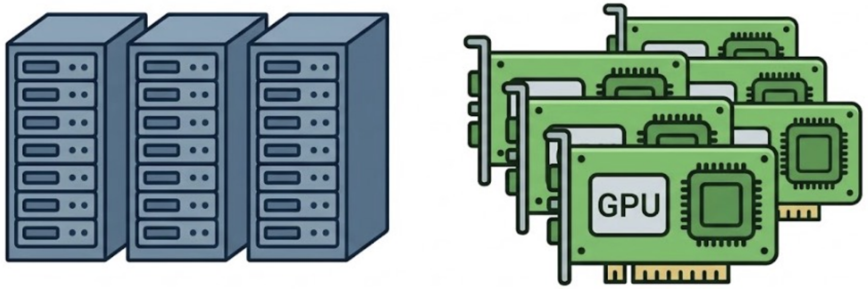
- Users upload code and launch the training job
- Cloud providers are responsible for reliable computation and networking
- Cloud providers typically lack access to the users' code

ASPLOS 2026

Anomalies in Training

High Frequency due to Scale & Time

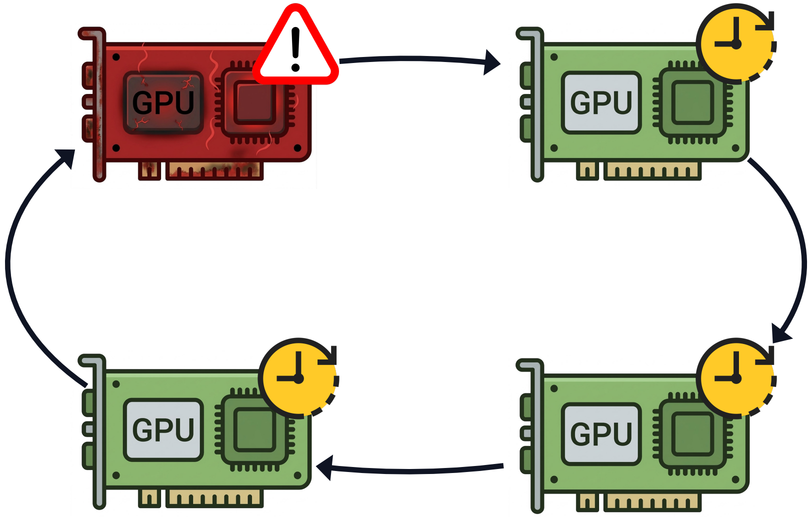
Thousands of GPUs



Months of training

Cascading Effect due to Synchronization

Single-node anomalies propagate rapidly



The high frequency and cascading effect of anomalies complicate diagnosis

Anomaly Taxonomy

Communication Anomaly

Fail-slow ⟷ Fail-stop

Communication Fail-slow
- PCIe contention
- Network congestion
- …

Communication Fail-stop
- NIC down
- Switch reboot
- …

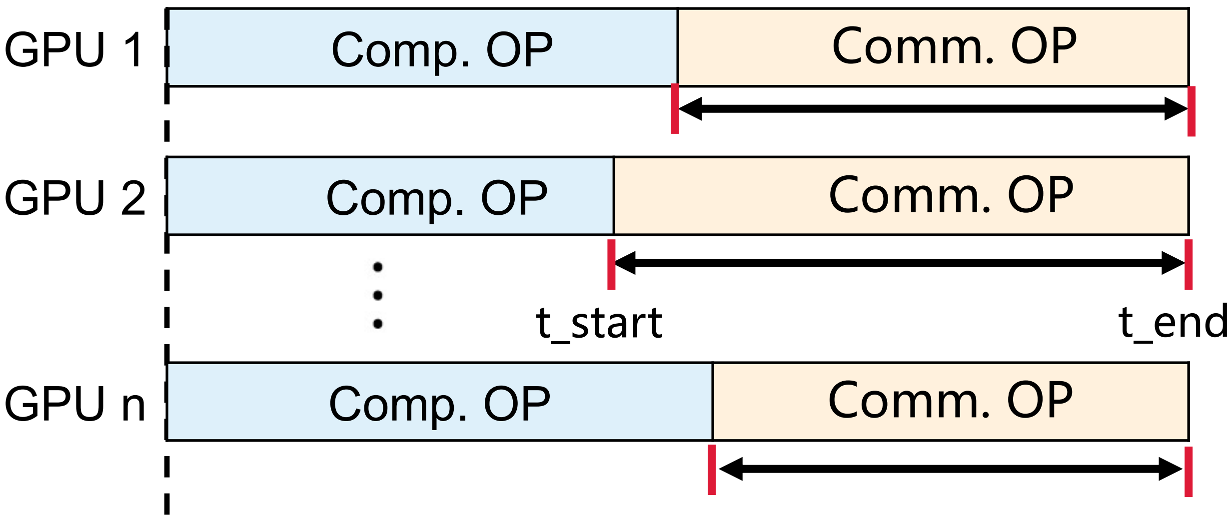
Computation Fail-slow
- GPU throttling
- CPU contention
- …

Computation Fail-stop
- GPU execution error
- CUDA error
- …

Computation Anomaly

Our goal: Detect and localize these anomalies accurately

Existing SOTA Monitoring Systems



SOTA methods (Aegis, NSDI 25; Holmes, NSDI 25; ...) log start and end timestamps of communication OP

- The start timestamp reflects the end of Comp. OP
- Localize anomalies via cross-node similarity

Pros

- **OP-level monitoring provides runtime visibility into training process**

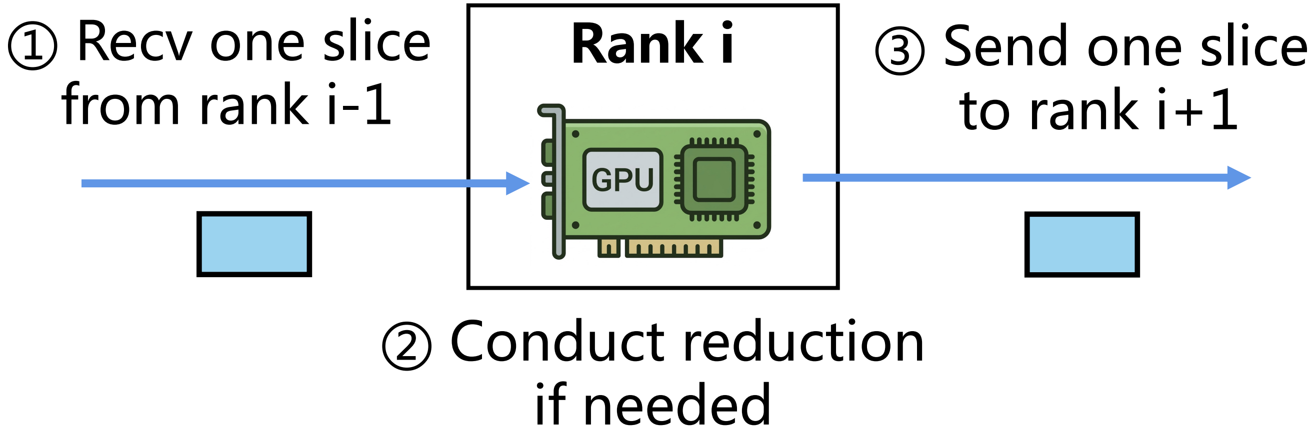
Cons

- **Too coarse-grained for accurate diagnosis**
- **Require intrusive code modifications**

OP-level Fails to Observe Transmission Gaps

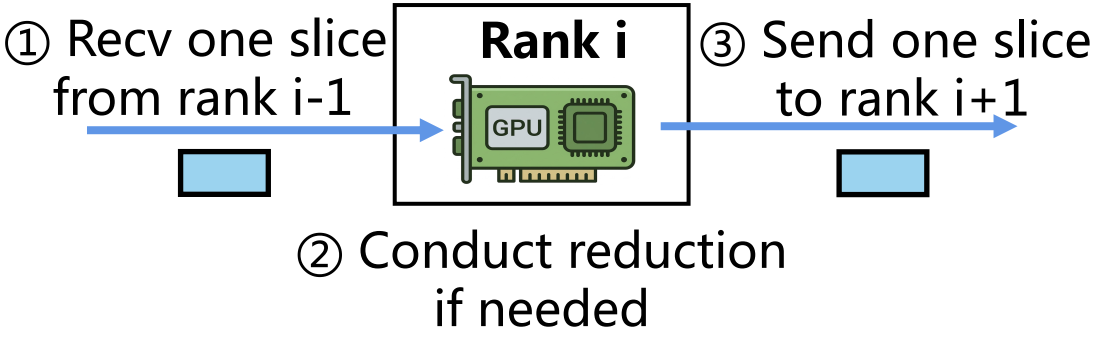
Data transmission and synchronization are performed at the **slice** level

- The default size of one slice is 1 MB
- Each rank receives one slice from its predecessor, and send to its successor



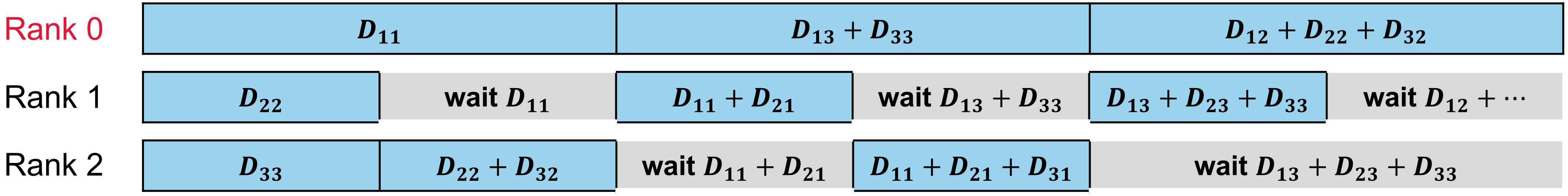
OP-level Fails to Observe Transmission Gaps

Data transmission and synchronization are performed at the slice level



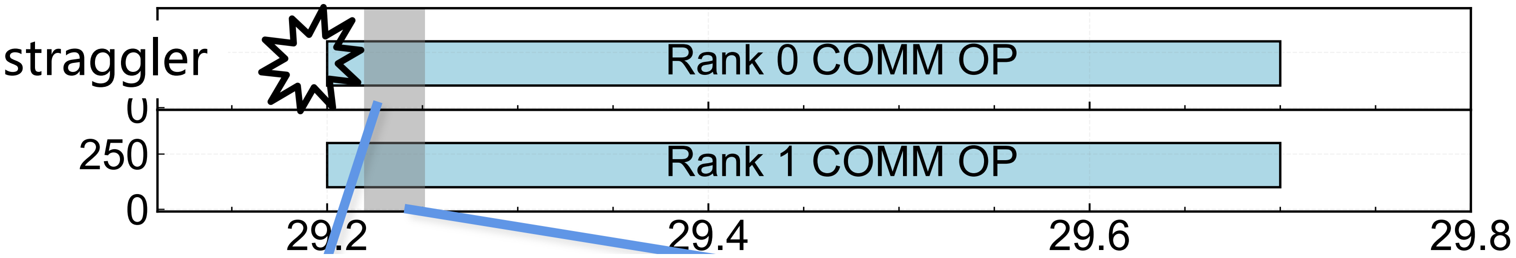
Stragglers cause transmission **µs-level** gaps on other ranks

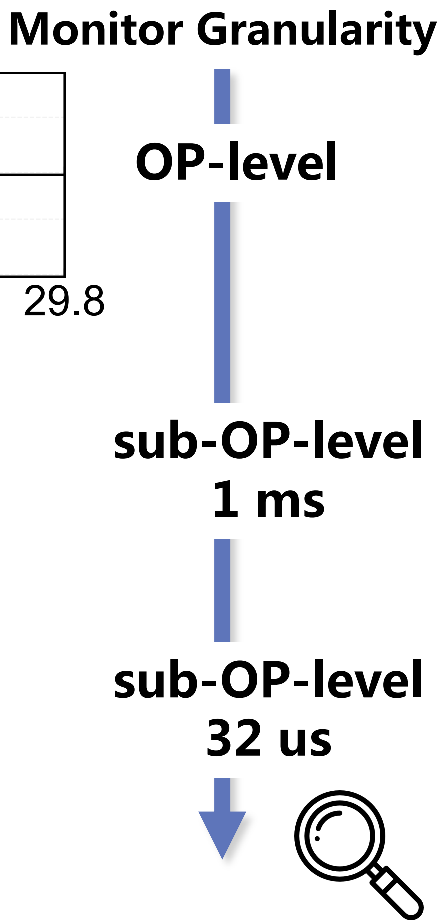
- Transferring 1 MB at 400 Gbps network takes about 20 us
- If stragglers' sending rate reduces to half, its successor need wait for 20 us, causing gaps



μs-level Transmission Gaps

4-node AllReduce with one rank's bandwidth reduced by 50%



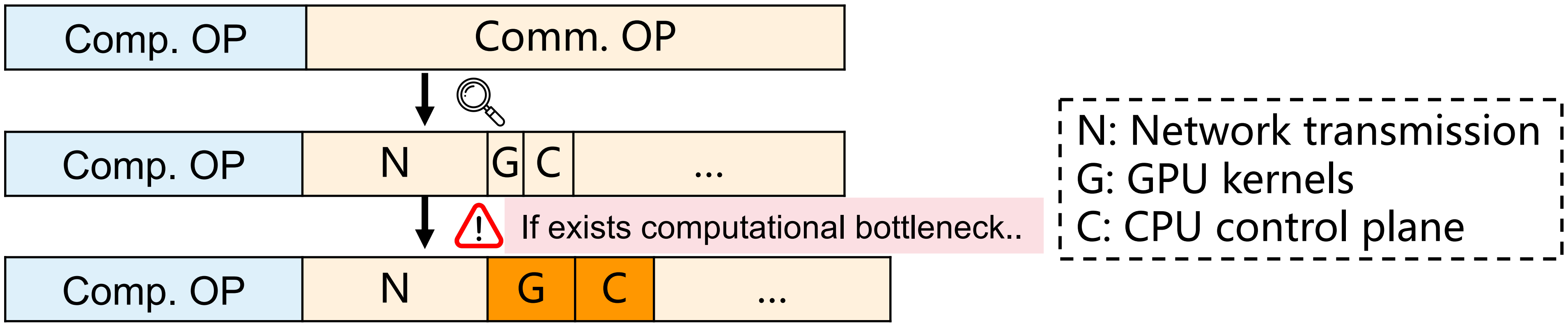


OP-level incorporates computation overhead

Collective communication is not simply network transmission

- Proxy threads serve as the control plane of GPU data transmission
- GPU kernels conduct reduction and copy data to buffers
- RNICs send/receive data

Computational bottleneck can also extend the duration of Comm. OPs

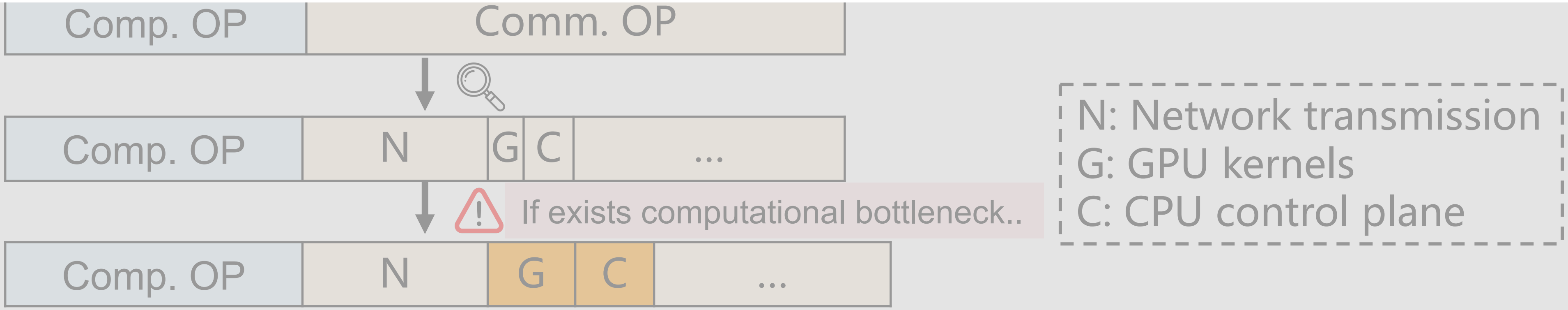


OP-level incorporates computation overhead

Collective communication is not simply network transmission.

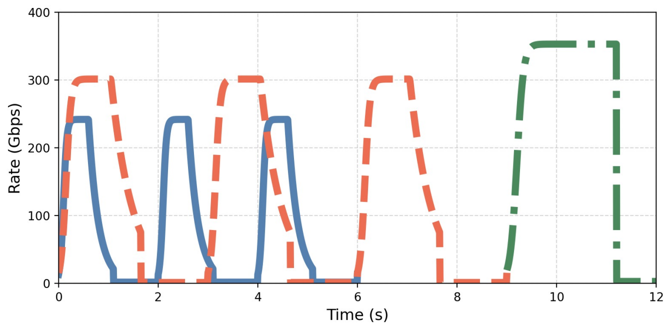
- Proxy threads serve as the control plane of GPU data transmission.

An ideal monitoring system for LLM training should offer fine-grained visibility while remain non-intrusive



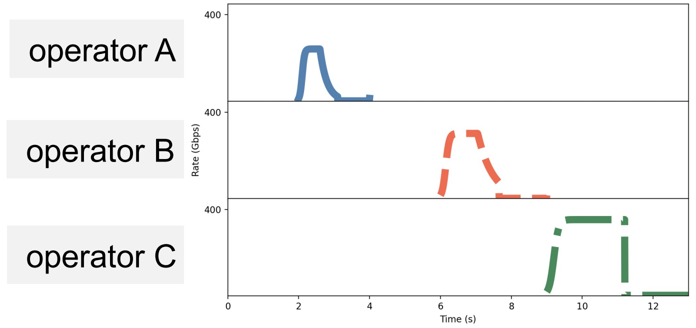
Key Idea of Pulse

Fine-grained and non-intrusive training monitoring via traffic measurement



µs-level per-flow measurement

🔗 flow-operator association



per-operator transmission rate

Fine-grained visibility

Provide sub-OP-level visibility into collective communication

Non-intrusive monitoring

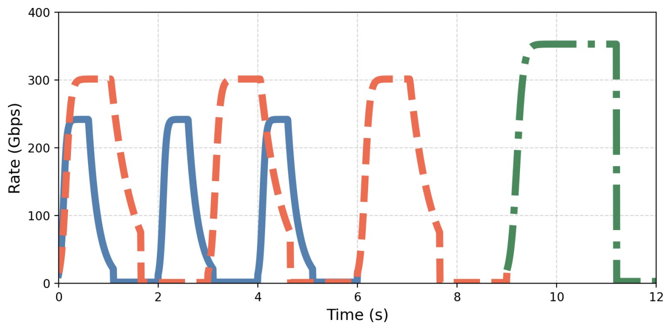
No instrumentation in training code or libraries

Minimal interference

Run in an isolated process or directly on hardware

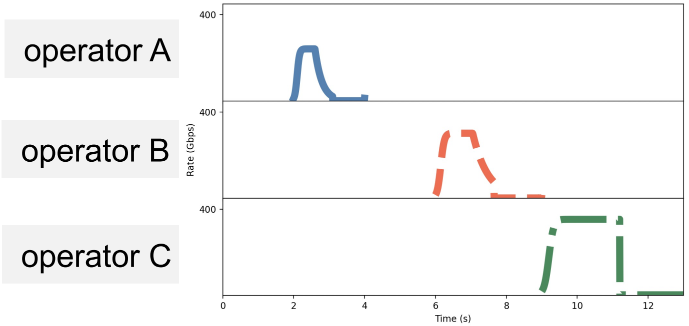
Design Challenges

Fine-grained and non-intrusive training monitoring via traffic measurement



µs-level per-flow measurement

🔗 flow-operator association



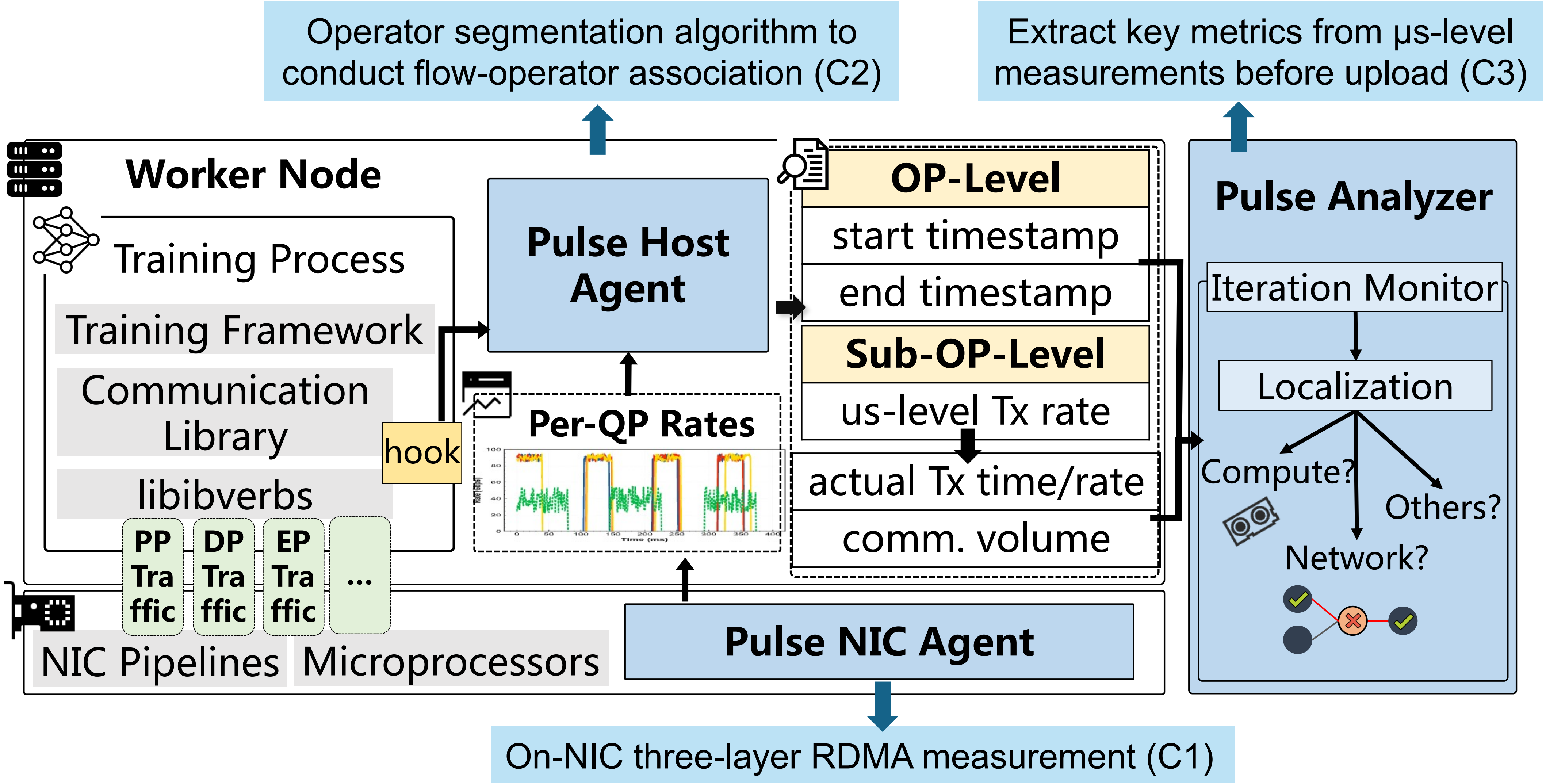
per-operator transmission rate

C1. How to conduct µs-level RDMA measurement

C2. How to bridge traffic layer with application layer

C3. How to handle massive µs-level monitoring data

Pulse Overview



RDMA Measurement on NIC Agent

Host-based measurement

- RDMA is fully offloaded to RNICs
- Lack actual transmission timing
- Leads to inaccurate measurement

P4-switch-based measurement

- Architecture limits the precision
- P4-compatible switch is not yet widely available

SmartNIC-based measurement

On-NIC microprocessors can access NIC events (*e.g.*, TX event)

Equipped with DRAMs or can access to host's memory

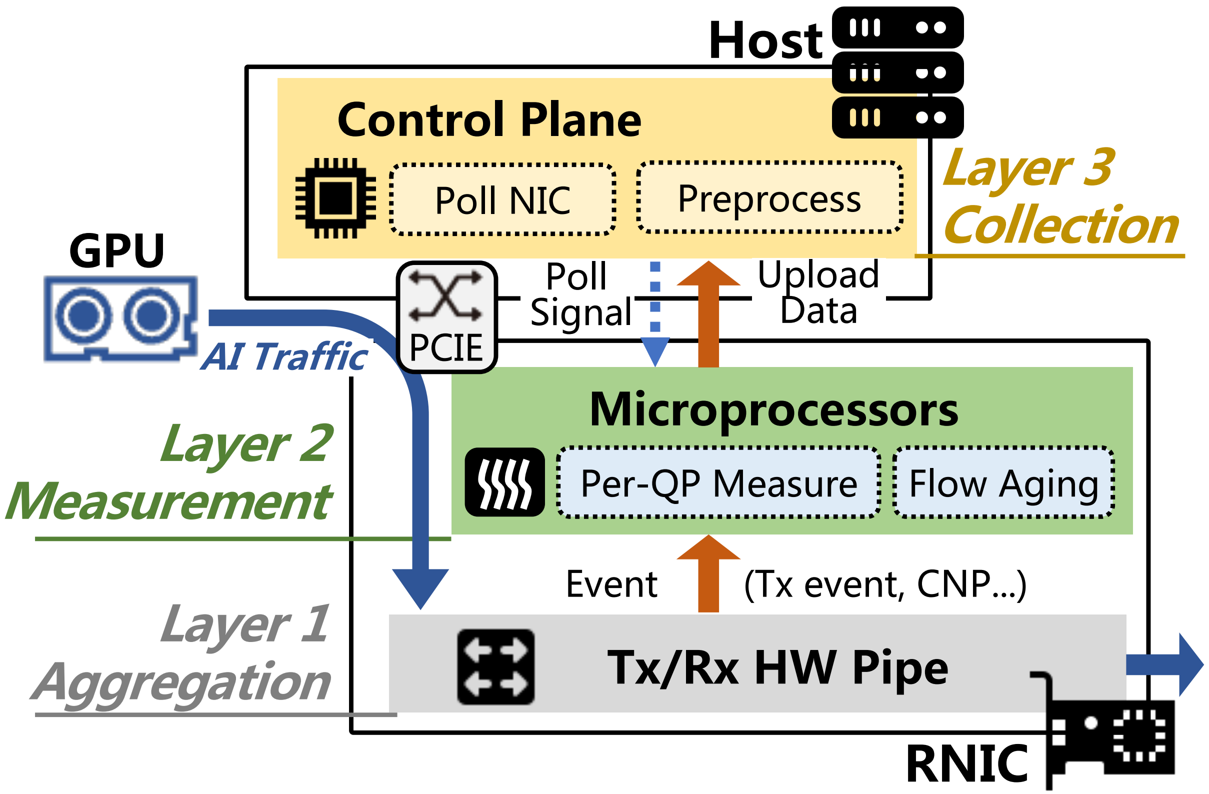
Follow RTC architecture with fewer constraints

Modern smartNICs provide a new RDMA measurement opportunity

On-NIC Three-Layer Measurement

Challenge: Per-packet processing places measurement on the packet-processing critical path

- **Aggregation layer** on HW pipeline
  - Trigger measurement events after transmitting a data chunk
- **Measurement layer** on microprocessors
  - Record per-QP transmitted bytes within each time window
- **Collection layer** on CPUs
  - Poll NIC data periodically and coordinate with **measurement layer** to reclaim inactive flows resources

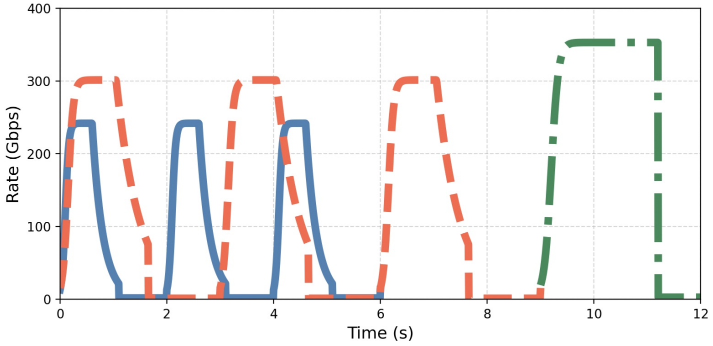


Supports µs-level measurement for up to 2,000 active RDMA flows per NIC

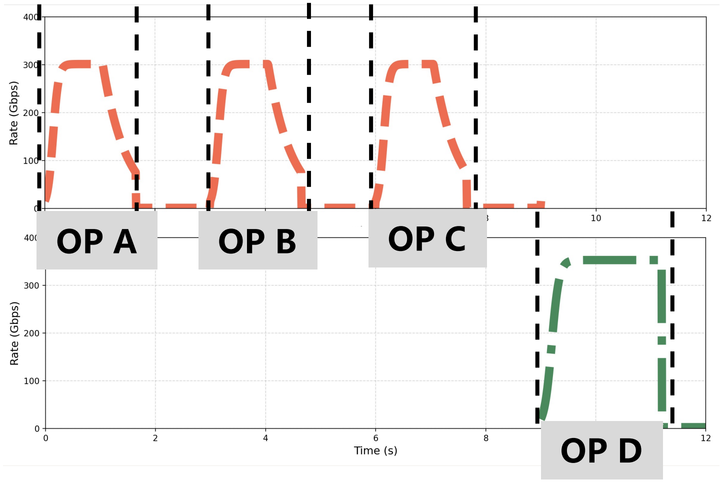
(see our paper for scalability analysis)

Flow-Operator Association

Goal: map each comm. operator to its corresponding flow measurement data



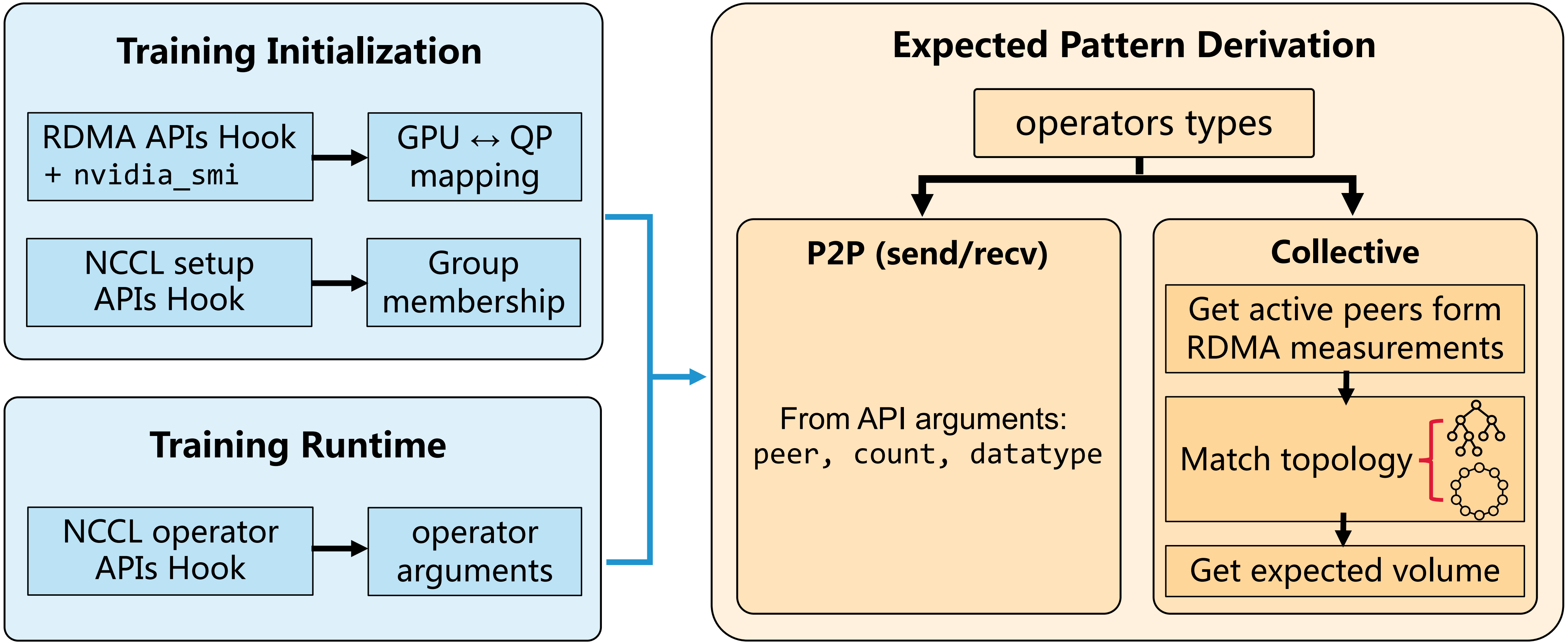
Identify **time boundaries** of each operator in the measurements



Key observation: Operators have predictable communication patterns

1. Derive each operator's expected communication pattern (peers and data volume)
2. Segment each peer's measurement data using the expected volume

Expected Volume and Peer Derivation



Leverage function hooking to avoid code modification.

Communication Operator Partition

Problem

Segmenting traffic using expected volume alone is unreliable

- additional synchronization messages
- packet headers add extra bytes
- retransmissions increase actual traffic

Observation

Expected volume provides a lower bound of actual traffic

Solution

Combine expected volume and time interval. Segment traffic when:

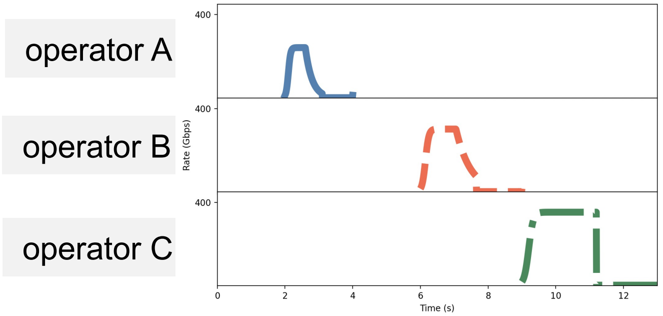
- transmitted volume ≥ expected volume
- idle interval $> 0(\mathrm{ms})$

After segmentation, Pulse obtains μs-level per-operator transmission rates

Fine-grained yet Lightweight Diagnosis

Problem: Uploading raw measurements incurs high overhead

Extract key metrics from raw measurements and only upload them



Per-operator transmission rate

- start, end timestamps → For computation anomaly localization
- actual communication time → Straggler shows large actual comm. time
  - sum of epochs with nonzero rate
- communication volume → Hung node sends the least data
  - total data sent by each node

Analyzer

Evaluation Setup

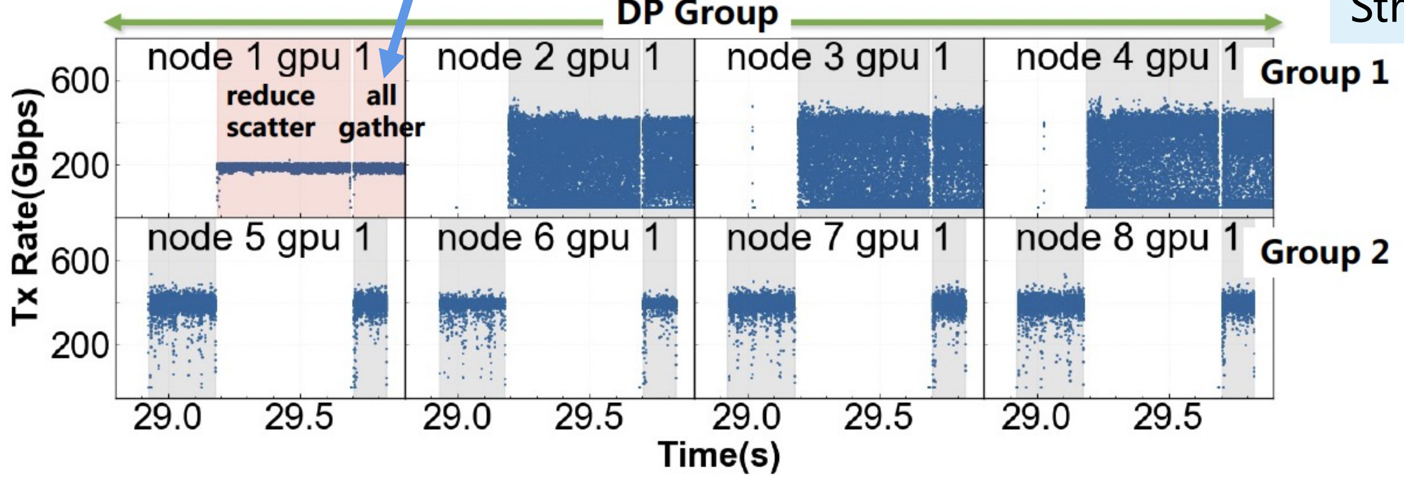
- Deploy Pulse on a 64 H200 GPUs cluster, and compare it with SOTA methods that rely on OP-level monitoring
  - Each server is equipped with 8 GPUs and 8 Bluefield-3 SuperNIC
  - All RNICs connect to one switch via 400 Gbps RoCEv2 links, with GPUDirect RDMA enabled
- RDMA measurement is implemented on Bluefield-3 SuperNIC using DOCA 2.9.0, while other parts are developed on host CPUs
  - Measurement time window (*i.e.*, epoch) set to 32 us
- Employ the Megatron training framework
  - Manually inject 12 common failure types during the training of GPT-2 and Mixtral 8×7B

Case Study: Network Congestion

Launching background RDMA traffic reduces one RNIC's sending bandwidth by half

- During the training of GPT-2 70B model, with 8-way TP, 2-way PP, and 4-way DP

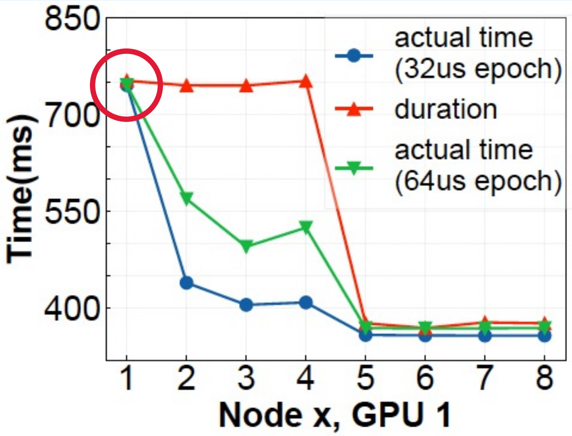
Straggler has a obviously low transmission rate.



Transmission rates of two DP groups

The entire group shows a long duration **(red line)**

Straggler has a longer actual comm. time **(blue line)**



Key performance metrics of two DP groups

Large time window folds some gaps into the actual comm. time **(green line)**

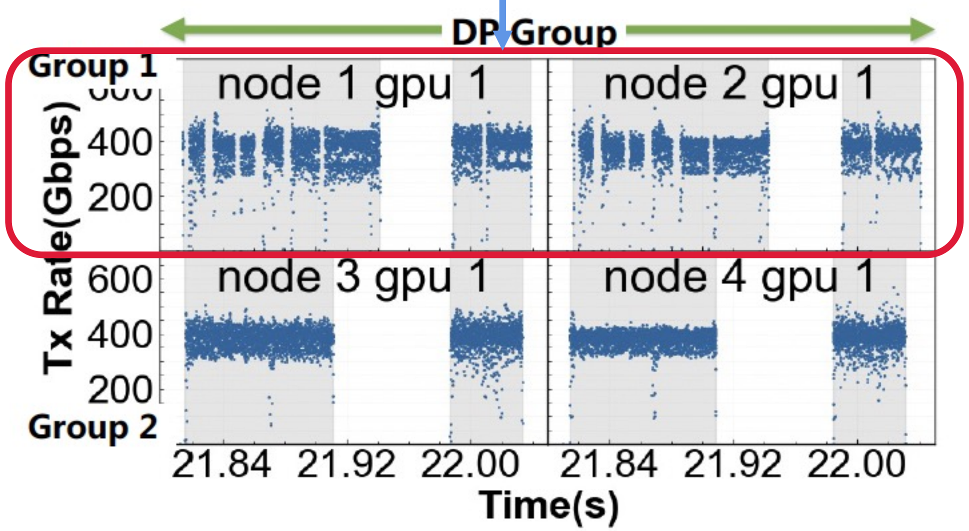
See our paper for the analysis of the required time window size

Case Study: CPU Contention

Use `stress` tool to saturate all CPU cores on one node

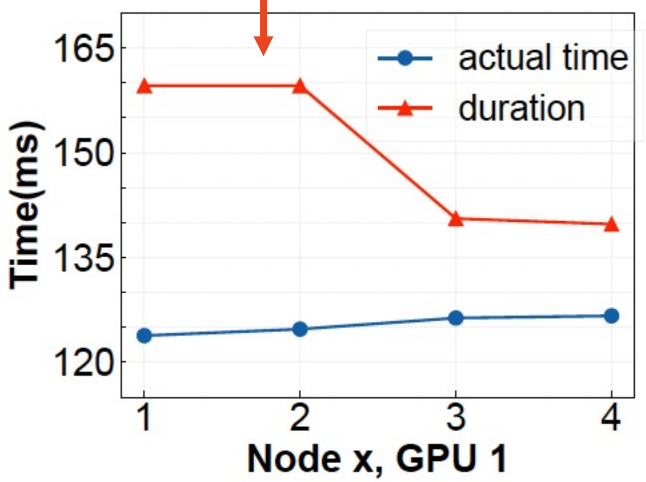
- During the training of GPT-2 32B model, with 8-way TP, 2-way PP, and 2-way DP

CPU contention prolongs the operator duration in Group 1



Transmission rates of two DP groups

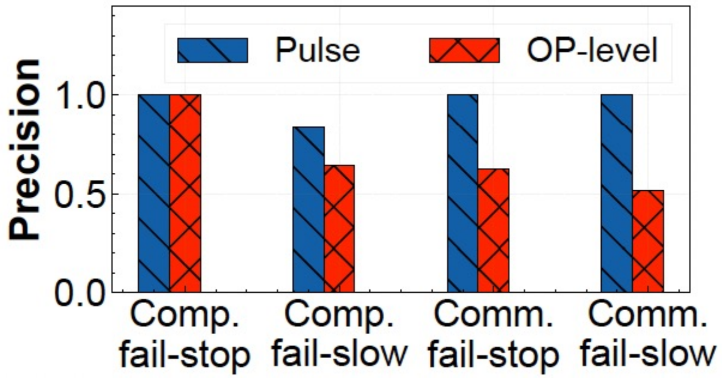
OP-level monitoring can misdiagnose this as network anomaly **(red line)**

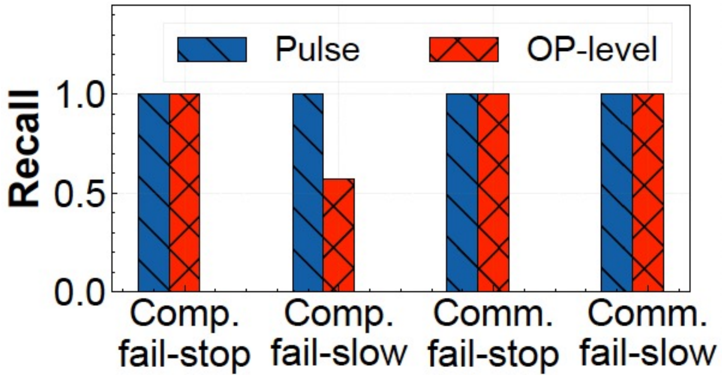


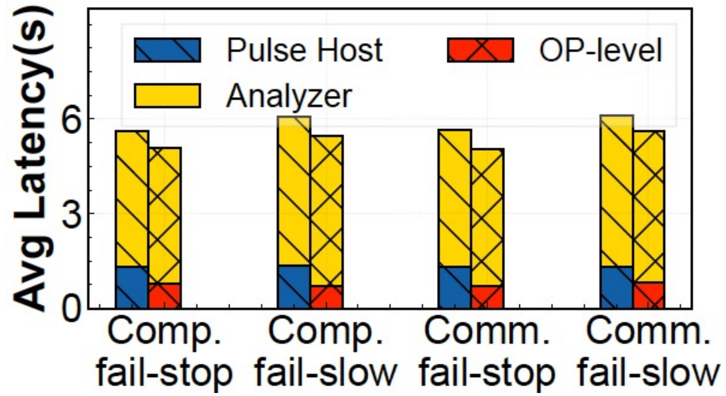
Key performance metrics of two DP groups

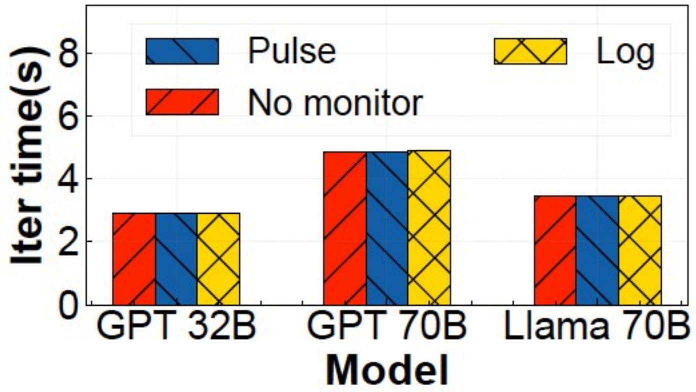
Identical actual comm. time across nodes indicates computational bottleneck during communication in Group 1 **(blue line)**

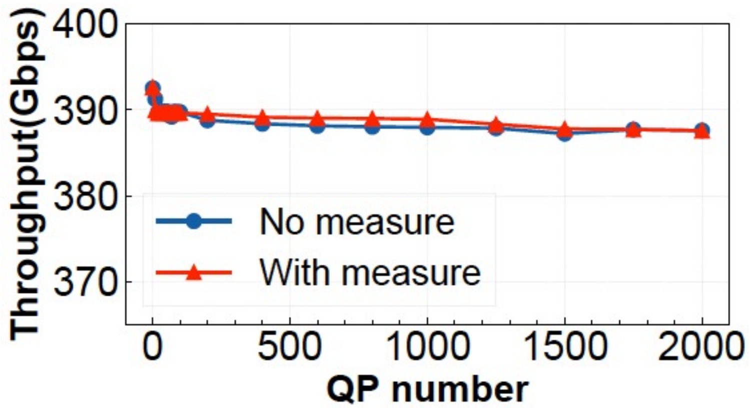
Evaluation Results

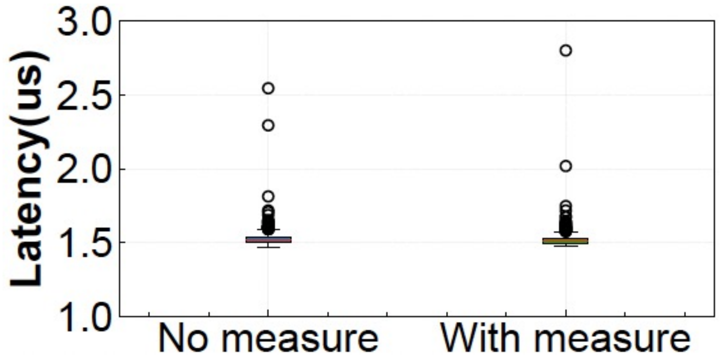












Pulse achieves high diagnosis accuracy and has minimal impact on training and RDAM performance

Conclusion

Pulse is a fine-grained and non-intrusive LLM training monitoring system.

- **Fine-grained**: Provide μs-level transmission rate monitoring of communication operators through traffic measurement.
- **Non-intrusive**: Leverage function hooking to help transform transport-layer measurement into operator measurements.

With fine-grained visibility, Pulse achieves machine-level anomaly localization for common failures, while imposing minimal impact on training performance.

Contact: yixiaonju@gmail.com

Thanks