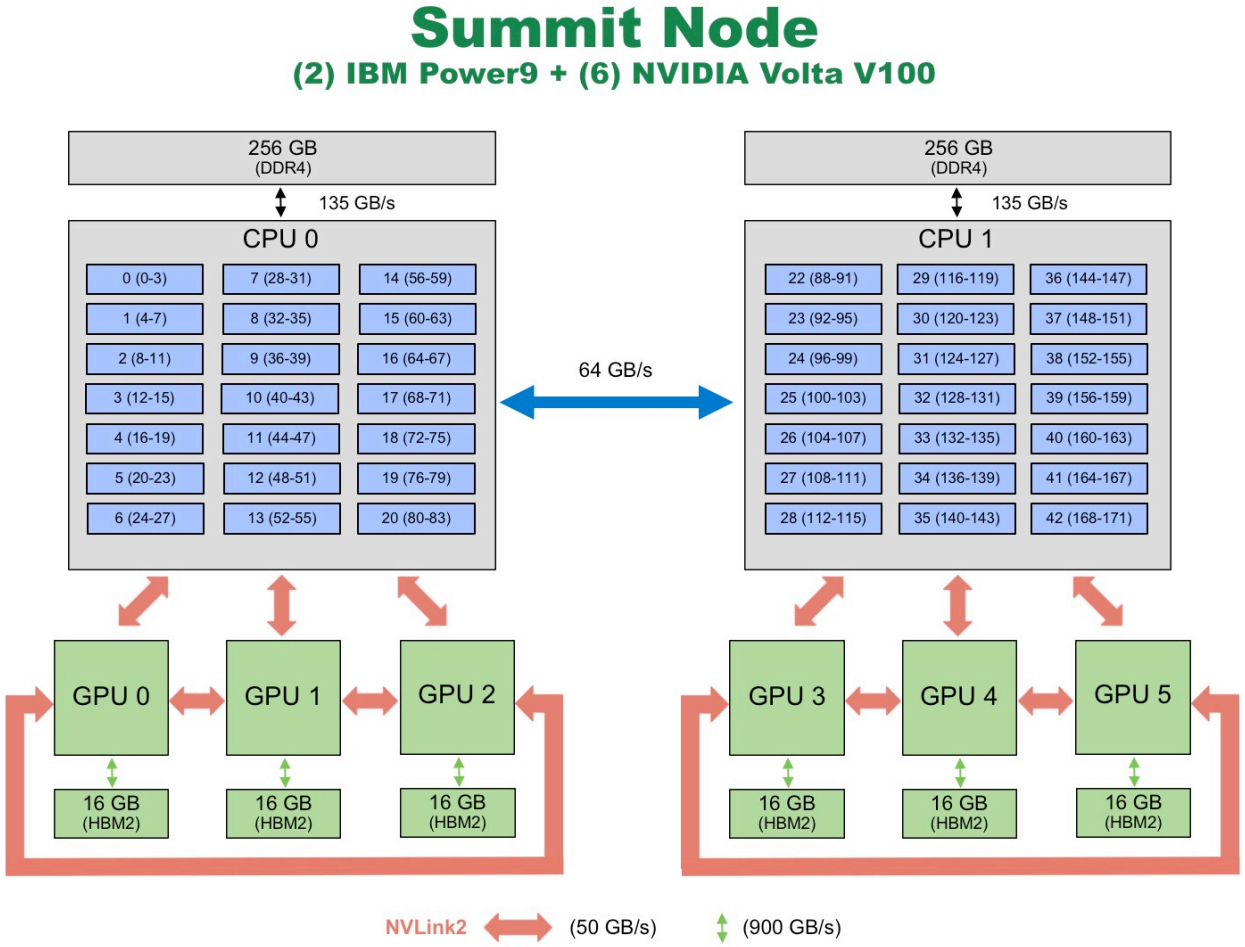
ncclsee: A Lightweight Profiling Tool for NCCL

Ioannis Vardas, Ruben Laso Rodriguez, and Majid Salimi Beni

Scaling AI: Distributed Training

- Training State-of-the-art AI models (especially LLMs) on a **single** GPU:
  - Too large to fit into the memory.
  - Slower training times.
- **Distributed training**: Facilitates such models by parallelizing the workload across **multiple** GPUs.
  - Either within a single machine or across multiple interconnected machines (nodes).
  - New challenge: **Communication** and co-operation of multiple GPUs.

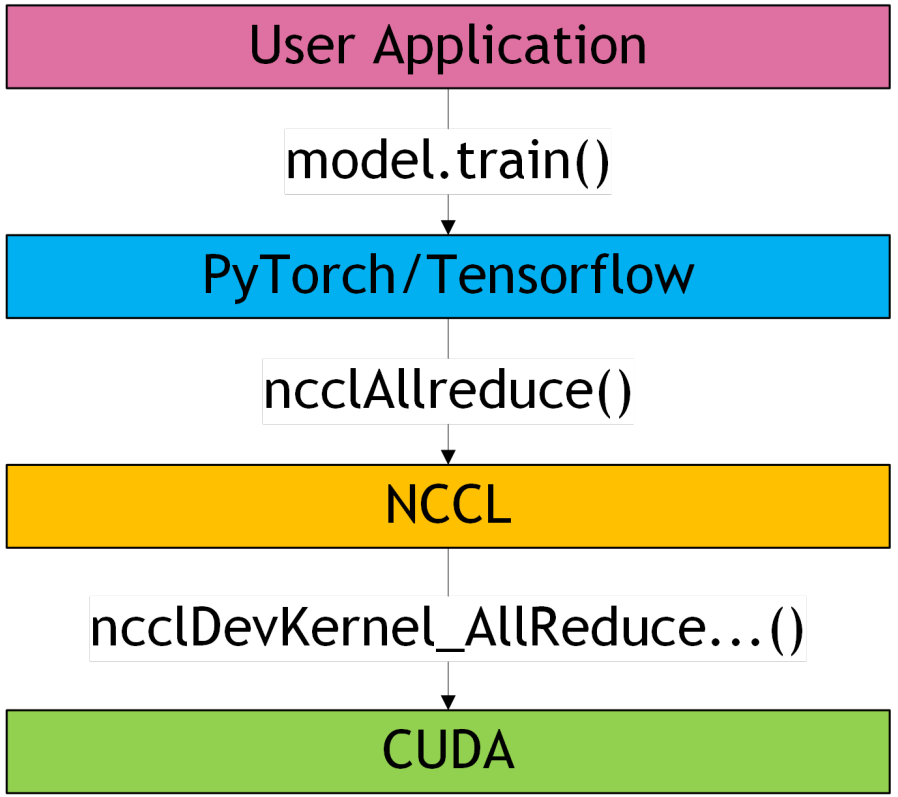


HTTPS://WWW.OLCF.ORNL.GOV/CALENDAR/PROGRAMMING-METHODS-FOR-SUMMITS-MULTI-GPU-NODES

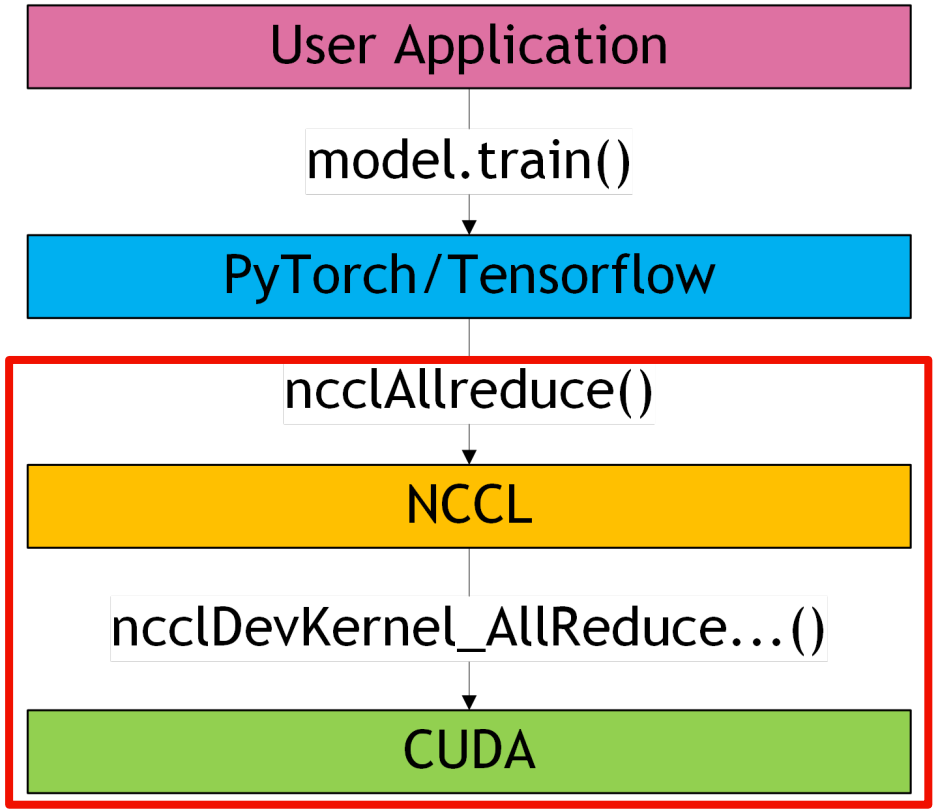
NCCL: Addressing the Inter-GPU Communication Challenge

- **NCCL**: NVIDIA Collective Communications Library.
  - Provides **collectives** optimized for NVIDIA GPUs and interconnects.
    - **Collective Operation:** A group of GPUs exchanges data and/or performs computations for a global result, e.g., **AllReduce, Broadcast.**
  - Performs such operations using optimized **CUDA** kernels.
    - Reduces memory copies.
    - Developers do not need to optimize their applications for specific machines.
  - GPU-communication backend for frameworks as PyTorch and TensorFlow.

NCCL in the AI Software Stack



NCCL in the AI Software Stack



The Challenge: Measuring NCCL Time at Scale

- Inter-GPU communication can impose significant overhead.
- To measure the communication time ➡ time NCCL operations.
  - Existing tool: NVIDIA Nsight Systems.
    - **Tracing**: produces a detailed event timeline.
    - Massive data files (prohibitive at large scales)
    - May introduce high overhead.
      - Distort execution.
  - For large scale runs **profiling** is more suitable:
    - Summarizes the time spent on different operations.
    - Concise, aggregated statistics.
    - Light overhead and smaller data files.

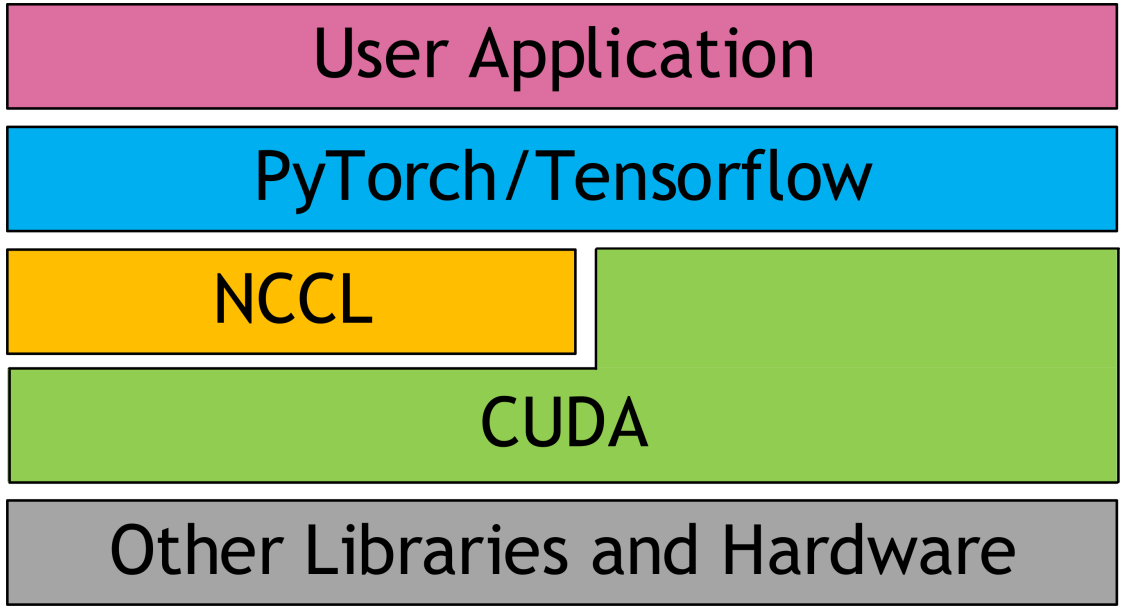
Introducing ncclsee: Lightweight NCCL Profiling

- Key Features & Approach:
  - **Profiling:** Aggregated statistics (call counts, average duration per NCCL operation).
    - Categorizes NCCL operations by buffer size (buckets).
  - **Low overhead**: Minimize impact on application performance.
    - Smaller data files suited for large-scale runs.
  - Interfaces with:
    - NCCL's built-in profiling interface.
    - CUPTI (CUDA Profiling Tools Interface) for timing CUDA kernels related to NCCL operations.

Profiling interfaces integrated with ncclsee

NCCL profiling interface

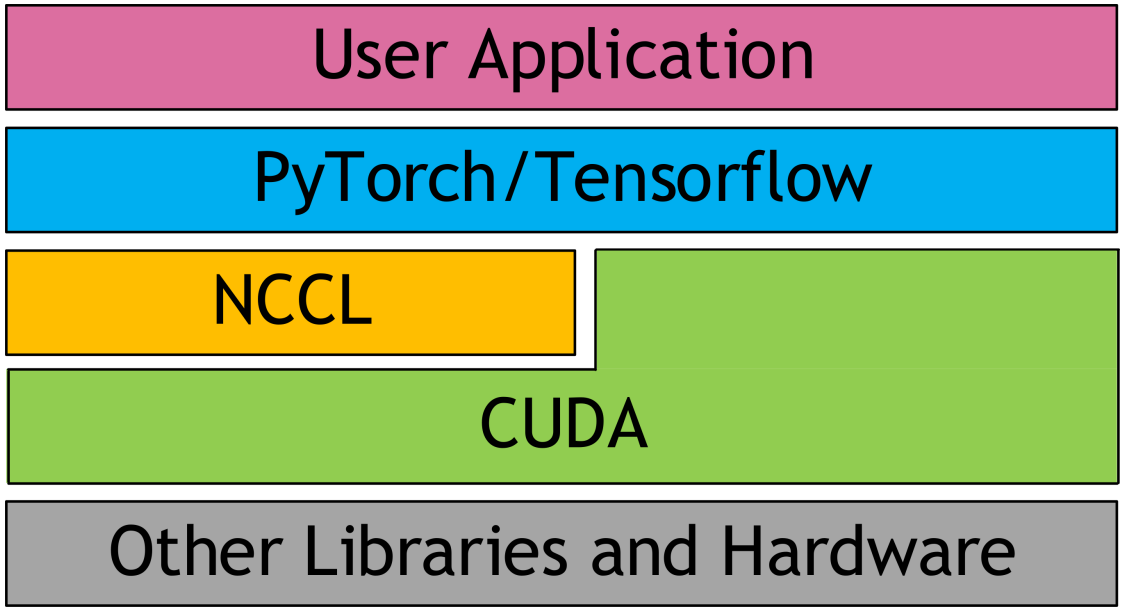
- NCCL operation
  - AllReduce, Broadcast, etc.
- Data size & type
  - Bytes transferred.
  - integer, float, bfloat16, etc.
- Asynchronous operations: Only provides the time until the NCCL operation is enqueued.
- **Missing:** The actual time to execute the operation (CUDA kernel).



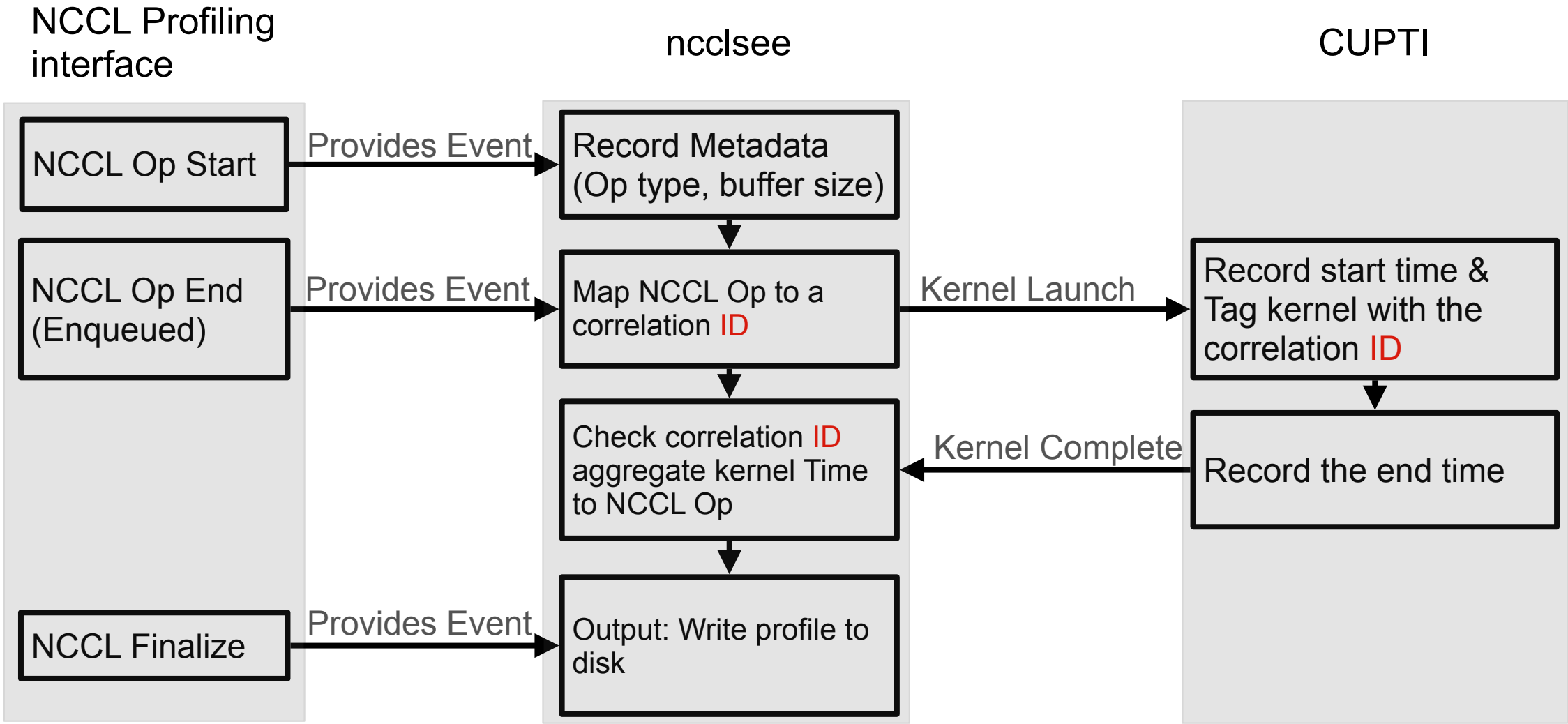
Profiling interfaces integrated with ncclsee

CUPTI: CUDA Profiling Tools Interface

- **Activity API**: Asynchronously record CUDA activities, e.g., Kernels, memory copies.
  - The **start** and **end** time of CUDA kernels.
- **External Correlation API**: Correlating NCCL operations with CUDA events

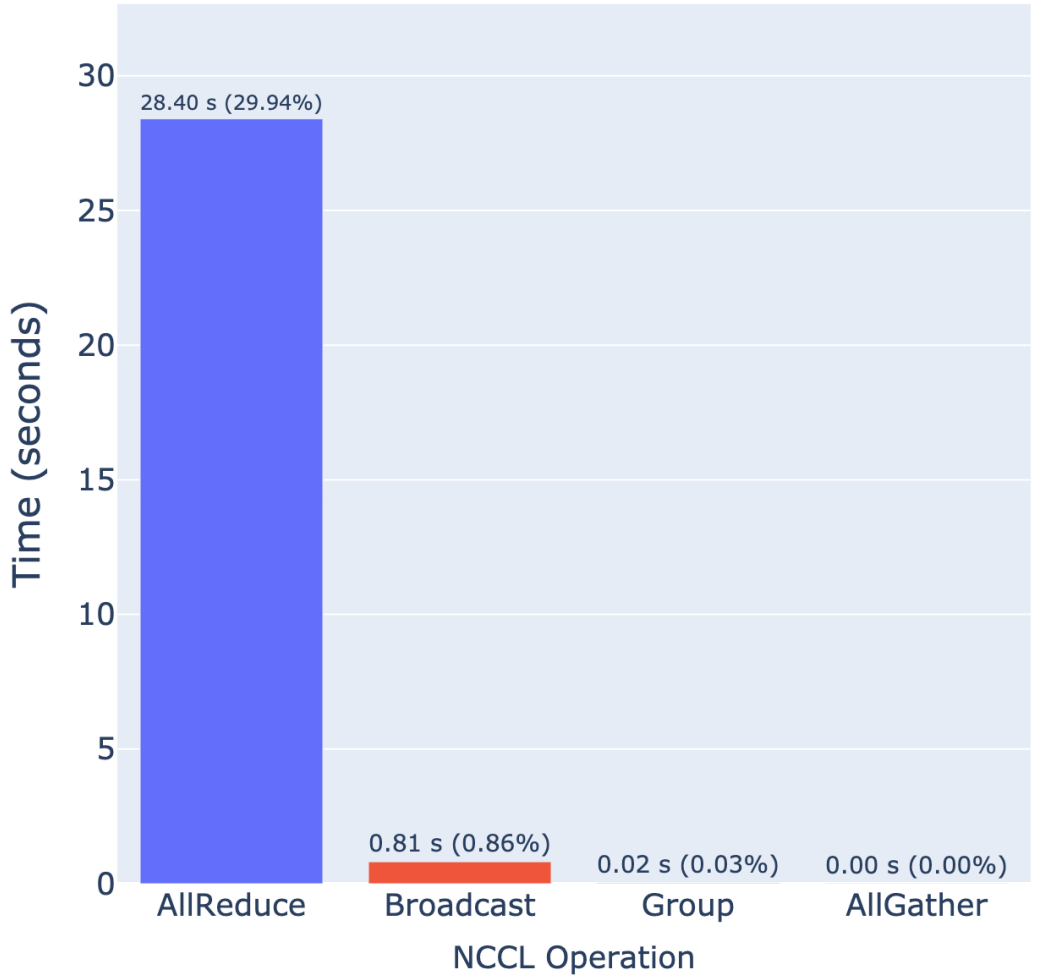


ncclsee: Combining NCCL profiling interface and CUPTI

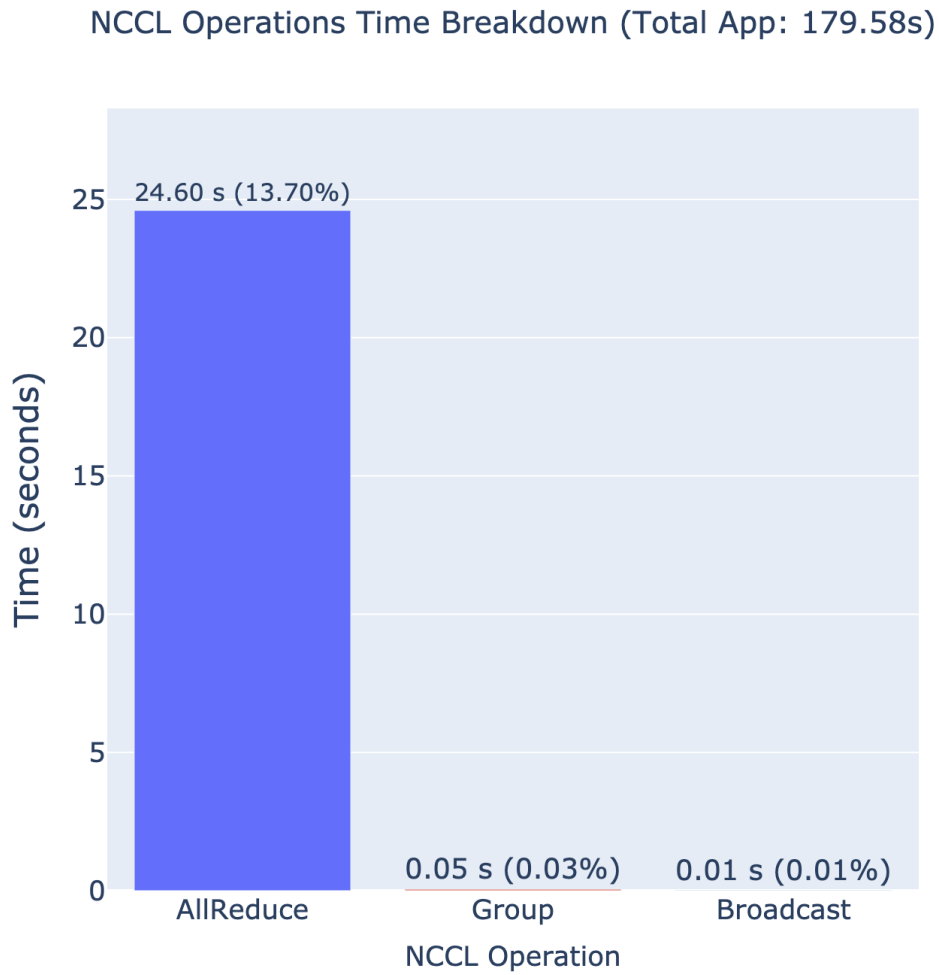


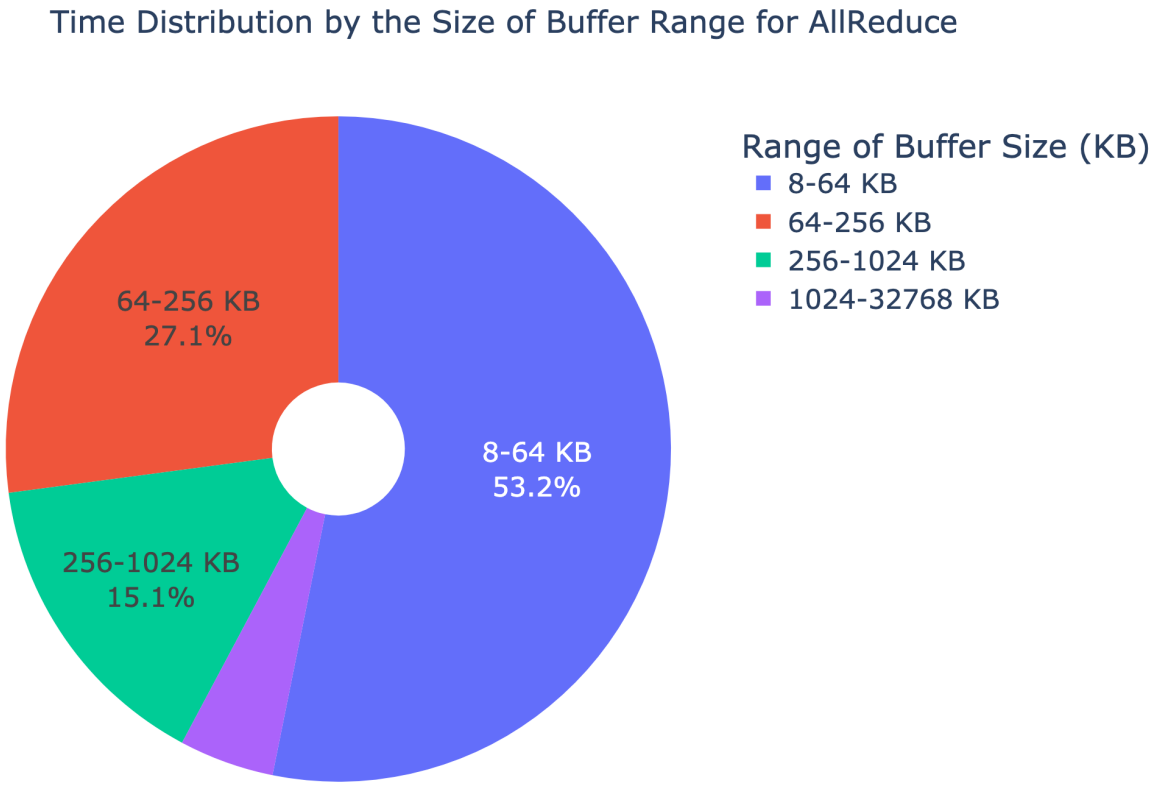
Profiling with ncclsee: ResNet50 training with PyTorch on two NVIDIA Tesla T4 GPUs

- Aggregate times among operations
- Despite only running with 2 GPUs, the time spent in NCCL operations is still significant.
  - **AllReduce** 30% of the total application time.
  - Broadcast is less than 1%.



Profiling with ncclsee: ResNet50 training with Tensorflow+horovod on two NVIDIA Tesla T4 GPUs





Summary

- Measuring Inter-GPU communication at scale is challenging.
  - **Tracing**: produces a detailed event timeline but higher overhead.
  - **Profiling**: more lightweight providing concise, aggregated statistics.
- **ncclsee** - a lightweight NCCL profiler:
  - Times NCCL collectives (incl. GPU kernels).
  - Statistics per range of buffer size.
- AI training:
  - Profile of ResNet50 training using either PyTorch or Tensorflow.
  - Identify optimization opportunities for NCCL operations.

Questions

Thank you!

ncclsee is under development and hosted on GitHub: https://github.com/variemai/ncclsee

NCCL on GitHub: https://github.com/NVIDIA/nccl

CUPTI: https://docs.nvidia.com/cupti/overview/overview.html