IOValve: Leakage-Free I/O Sandbox for Large-Scale Untrusted Data Processing

Sangho Lee, Jules Drean[*], Yue Tan[†], and Marcus Peinado

ACM Conf. Computer and Communications Security (CCS)

October 16, 2025

Microsoft

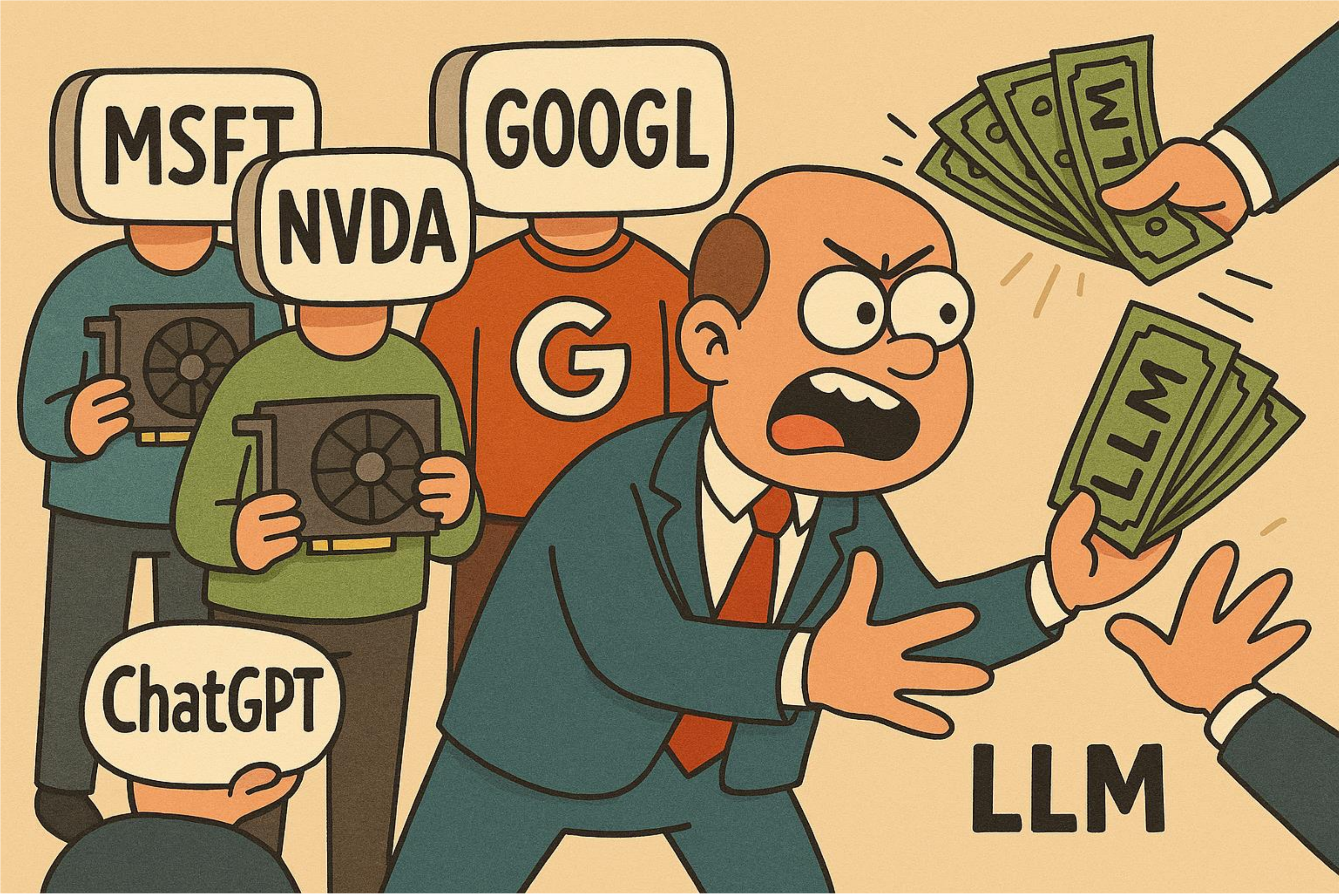
[*] Massachusetts Institute of Technology

Tinfoil

[†] Princeton University

Google

We, if not all, benefit from LLMs



deepseek

Claude

perplexity

Copilot

ChatGPT

Gemini

Can we use LLM in a confidential manner?

- Currently, our LLM usage relies on others without full control of our data.
  - **Software** (or service) to train and use LLMs doesn't belong to us.
  - **Hardware** to run the software with models as well.

🙋 Why not (securely) self-host or on-prem everything?

* https://help.openai.c… articles/7… data-usage-for-consumer-services-faq

LLM software stack is untrusted due to complexity and dependency

- OS: and drivers
- GPU API/runtime: **CUDA**
- Multi-GPU support: **NCCL**
- ML framework:
- Cluster management: slurm workload manager
- LLM serving:
- Base model: LLama
- ...

Oops! It's a kernel stack use-after-free: Exploiting NVIDIA's GPU Linux drivers

Posted Tue 14 October 2025
Author Robin Bastide
Category Exploitation
Tags exploitation, vulnerability, NVIDIA, GPU, CVE-2025-23280, CVE-2025-23330, 2025

Compromised PyTorch-nightly dependency chain between December 25th and December 30th, 2022.

By PyTorch Foundation | December 31, 2022

Probllama: Ollama Remote Code Execution Vulnerability (CVE-2024-37032) – Overview and Mitigations

Wiz Research discovered CVE-2024-37032, an easy-to-exploit Remote Code Execution vulnerability in the open-source AI Infrastructure project Ollama.

Sagi Tzadik
June 24, 2024
7 minute read

Inevitable hardware outsourcing due to massive computing power demand

Training Memory Requirements

The following table outlines the approximate memory requirements for training Llama 3.1 models using different techniques:

Model Size	Full Fine-tuning	LoRA	Q-LoRA
8B	60 GB	16 GB	6 GB
70B	500 GB	160 GB	48 GB
405B	3.25 TB	950 GB	250 GB

≥41 NVIDIA H100 GPUs 😱

Note: These are estimated values and may vary based on specific implementation details and optimizations.

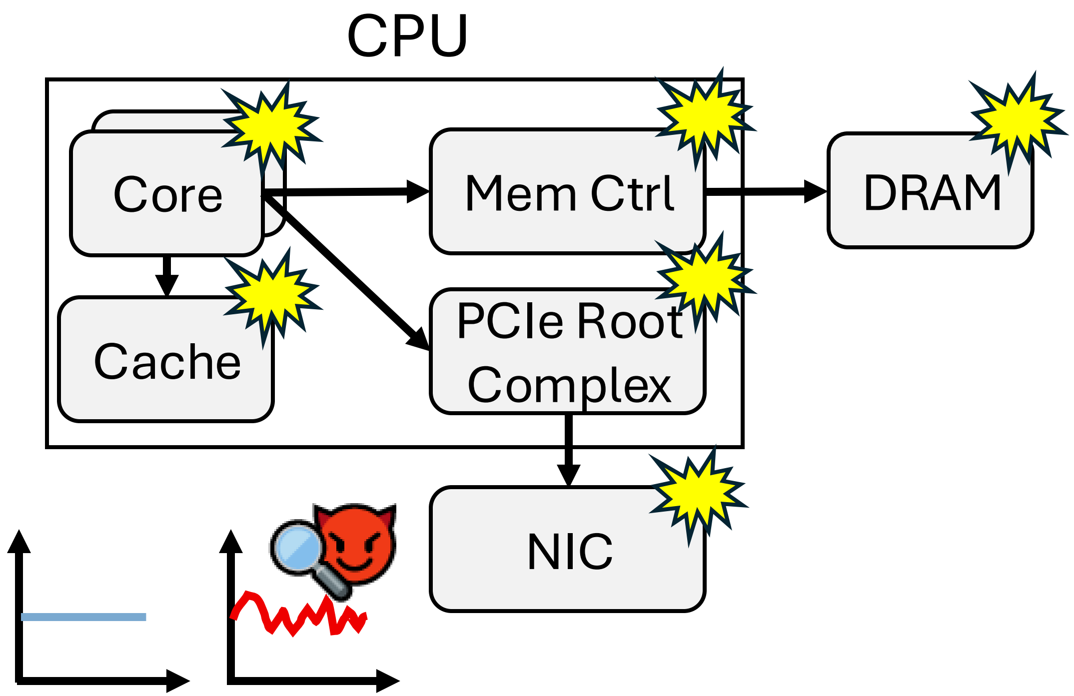
* P. Schmid et al., "Llama 3.1 - 405B, 70B & 8B with multilinguality and long context, " https://huggingface.co/blog/llama31, July 2024

What about sandbox + confidential cloud?

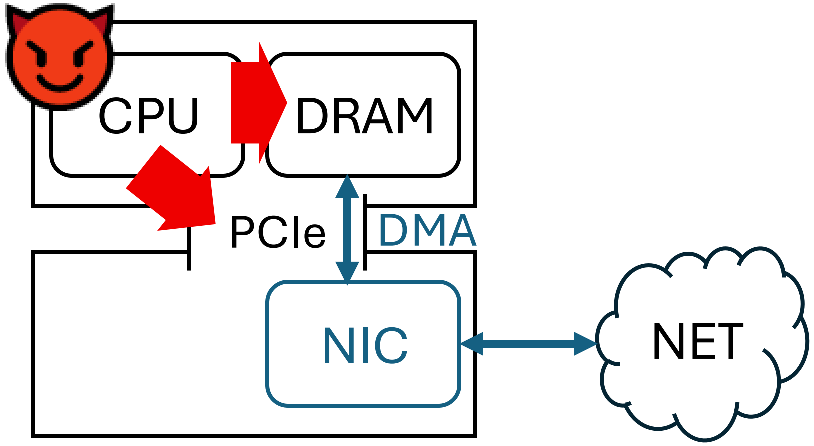
- Sandbox can confine untrusted software's behaviors
  - System call filter, IP packet filter, ...
- Confidential computing in the cloud prevents providers from accessing and affecting computation
  - Memory encryption and integrity, remote attestation, ...
- Best of both worlds. Problem solved

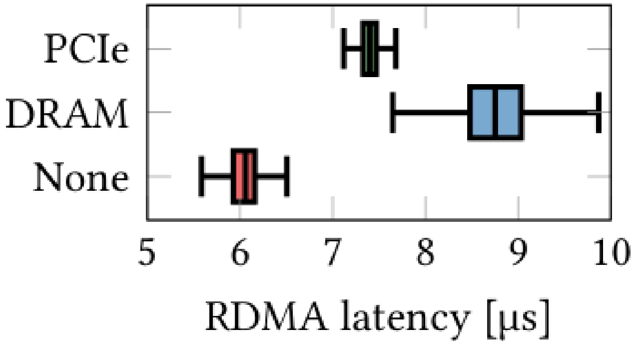
HW resource sharing → side/covert channels

- Existing techniques let adversary and victim share (some) HW resources.
  - CPU core and cache
  - Memory controller and DRAM
  - PCIe root complex and peripherals…
- Adversary can modulate data onto outgoing traffic by affecting its performance.



Host interconnect congestion does matter





Distinguishable
(two sample KS test: ≈1)

* S. Agarwal et al, “Understanding host interconnect congestion,” ACM HotNets 2022.

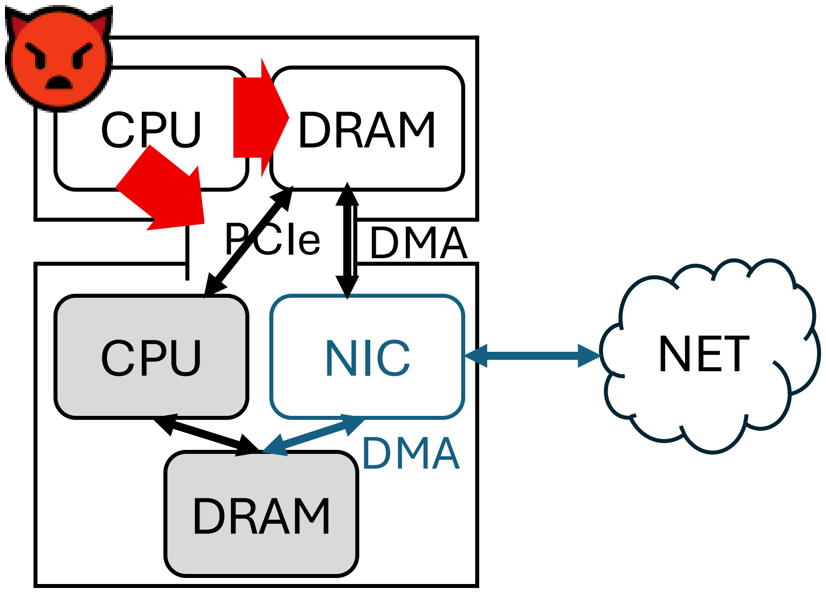
Rethink sandbox boundary: Physical host

- Step back from hardware resource sharing
- Draw the sandbox boundary around a physical host
  - Each tenant solely owns a host.
- Practical use case: Large-scale computation for LLM and others
  - Some tenants even require multiple hosts. No need to share resources.

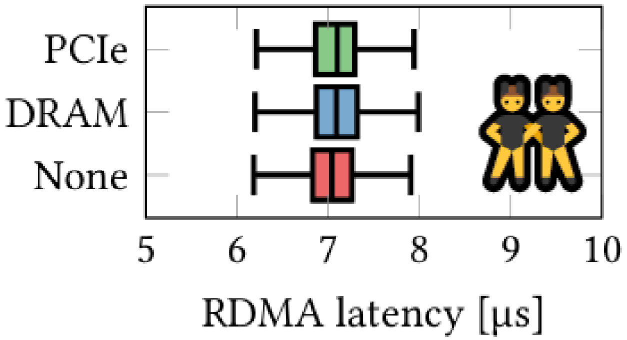
IOValve: I/O sandbox with physical isolation

- Sandbox boundary: physical host to run untrusted software
- I/O monitor: Physically separated HW to confine the host’s every network I/O activity
- Congestion-free memory transfer: The physical host and I/O monitor share nothing but minimal facilities to relay data.
- Traffic regularization: uniformize and encrypt every outgoing message

Congestion-free memory transfer



I/O Monitor



Indistinguishable
(two sample KS test: ≈0.07)

Data processing unit (DPU) as I/O monitor

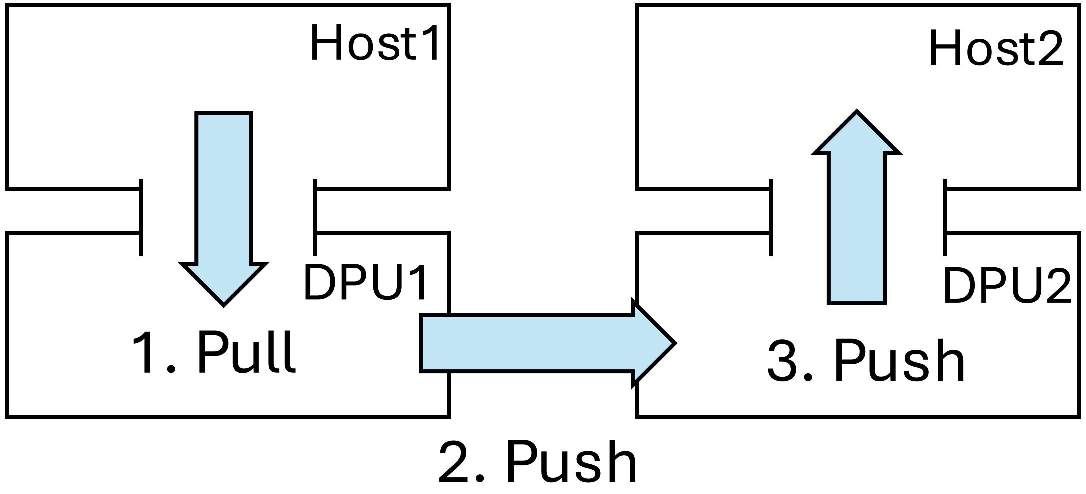
- Powerful single-board computer at PCIe
- General purpose computation: 10s of Arm cores, 10s of GBs of DRAM, NVMe SSD, ...
- Network acceleration: several 100s of Gbit/s RDMA/InfiniBand, line-rate encryption, Regex, ...
- Modern datacenters already feature DPUs.

NVIDIA BlueField-3 DPU

* https://marketplace.nvidia.com/en-us/enterprise/networking/bluefield3/

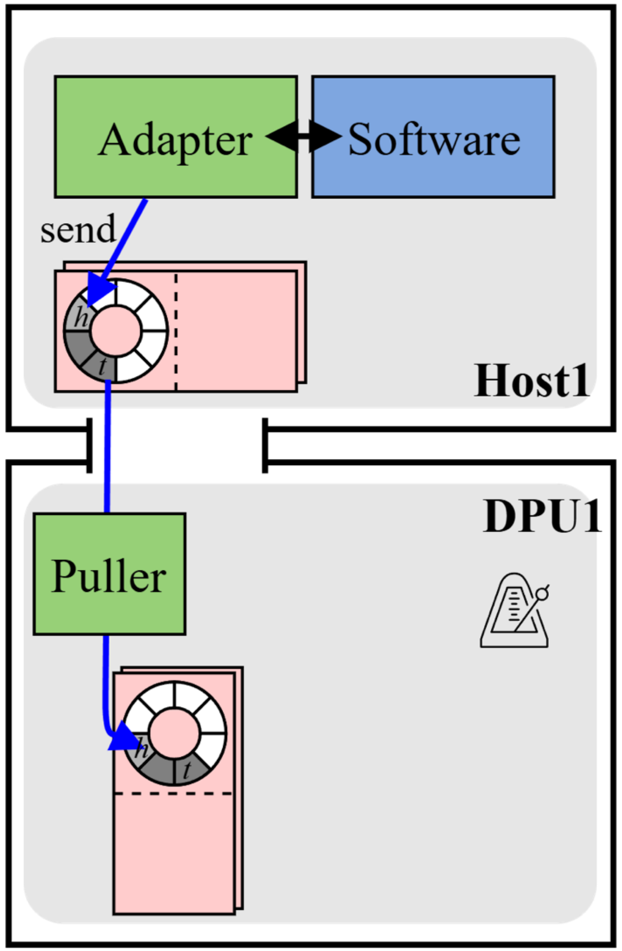
Three-stage unidirectional data transfer

- IOValve DPUs repetitively perform the following operations:

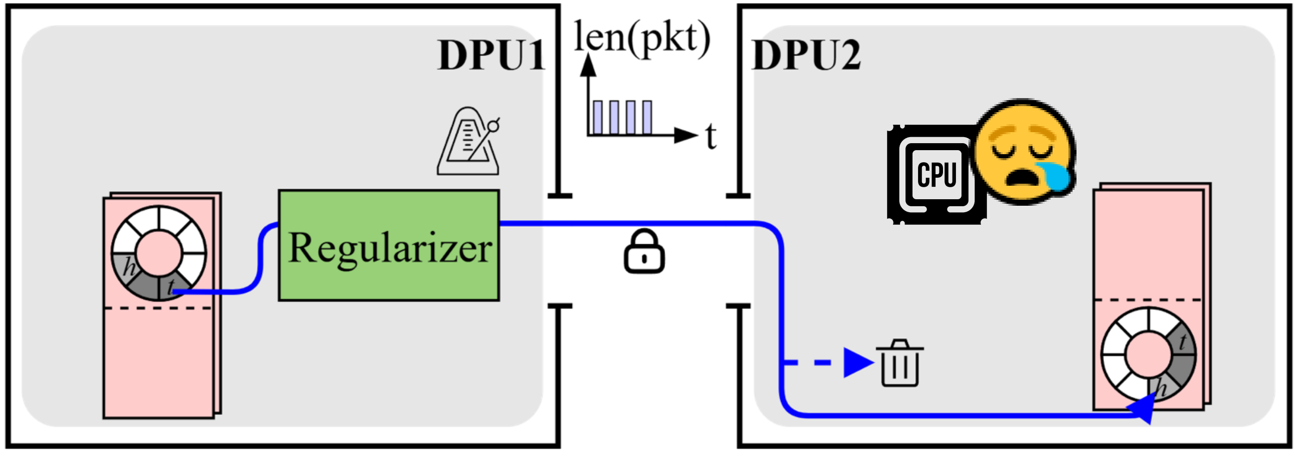


Pull data from host to DPU

- Periodically poll the host's send ring buffer
- Copy **fixed-size data** from host to DPU if
  - The host buffer has a new data item.
  - The DPU buffer has an empty slot for it.
- Use DPU's **DMA engine** for both operations, bypassing the CPU



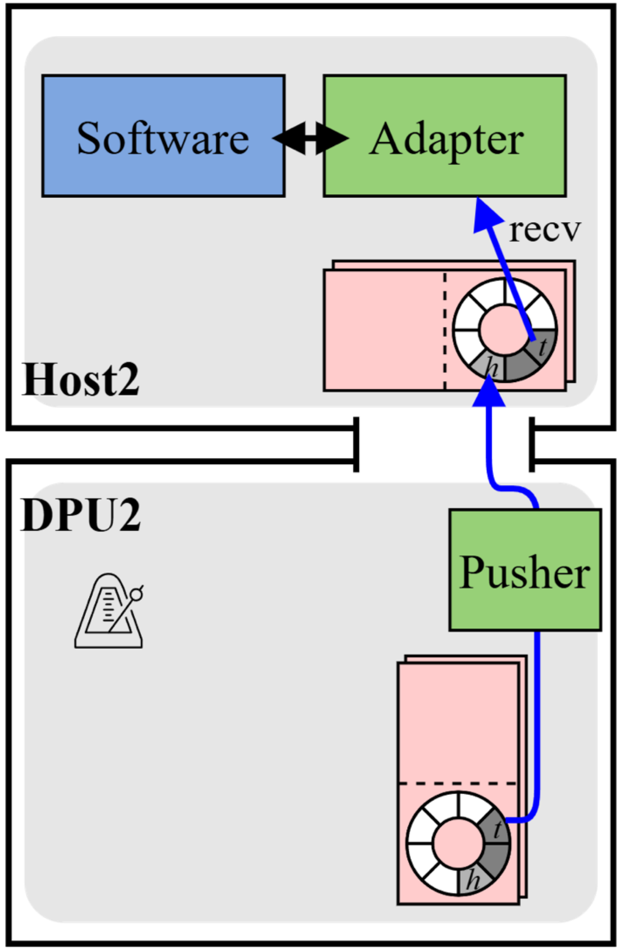
Push data to remote DPU



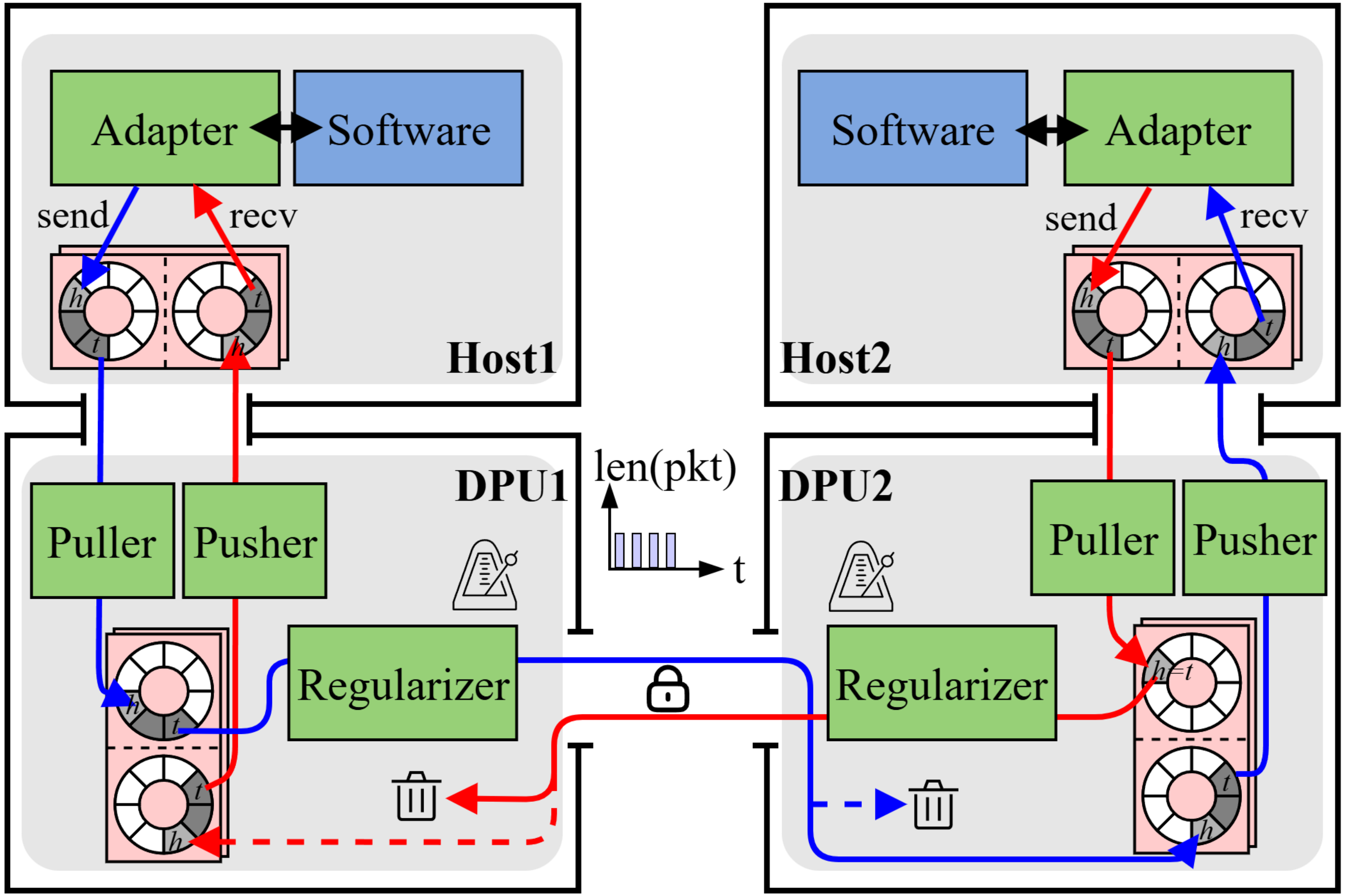
- Periodically poll the peer DPU's ring buffer
- Transfer **fixed-size real or fake data** to the peer
  - Real one if it has a new data item and the peer has an empty slot
  - Fake one if it has no data to send or the peer's buffer is full
- Use **one-sided RDMA READ/WRITE** to bypass the peer's CPU
- Encrypt RDMA over UDP (RoCEv2) with IPSec HW accelerator

Push data from DPU to host

- Periodically poll the host ring buffer
- Copy **fixed-size data** from DPU to host if
  - The DPU buffer has a new data item.
  - The host buffer has an empty slot for it.



Combined view of data exchange b/w hosts

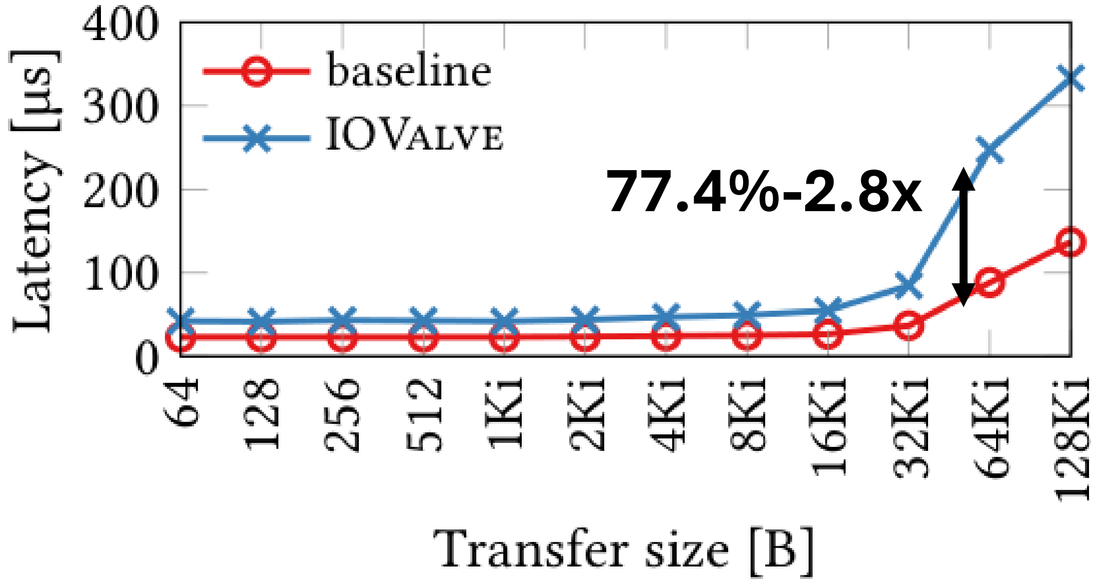


Evaluation setup

- Two nodes featuring
  - Two Intel Xeon Silver 4510 CPUs (24 cores at 2.4 GHz)
  - 256 GiB of RAM
  - 1 TB of NVMe SSD
  - NVIDIA BlueField-3 DPU with two 200 Gbit/s ports
  - NVIDIA TESLA P40 GPU (24 GiB of RAM)
- Connected with a **100 Gbit/s** cable

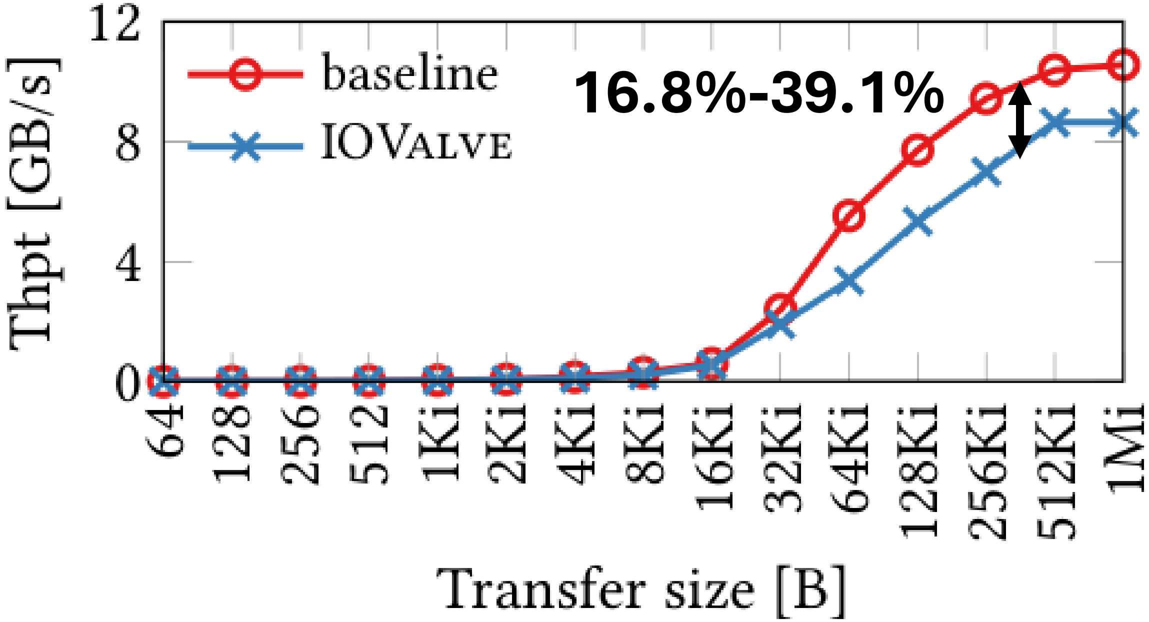
Evaluation: Latency and throughput

Latency



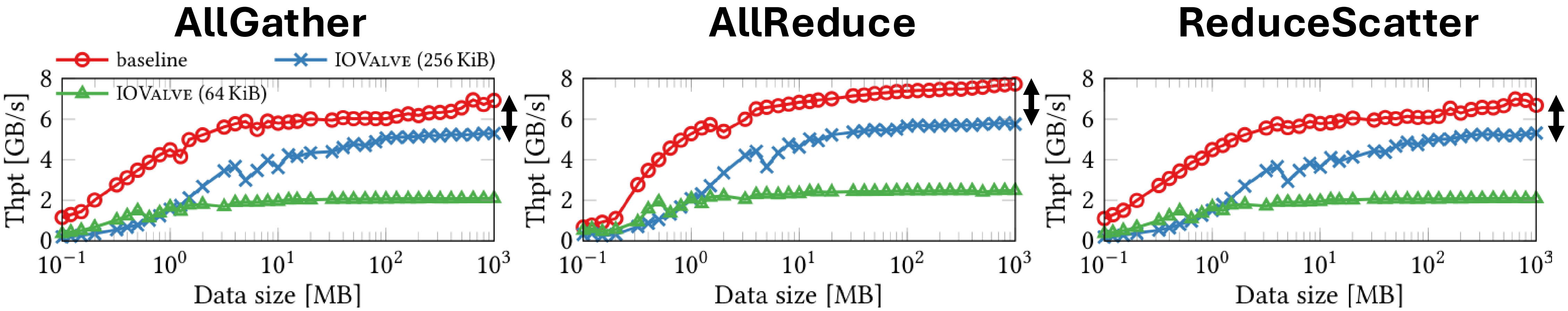
Expected latency overhead:
Single hop → three hops

Throughput



Benefit from pipelining

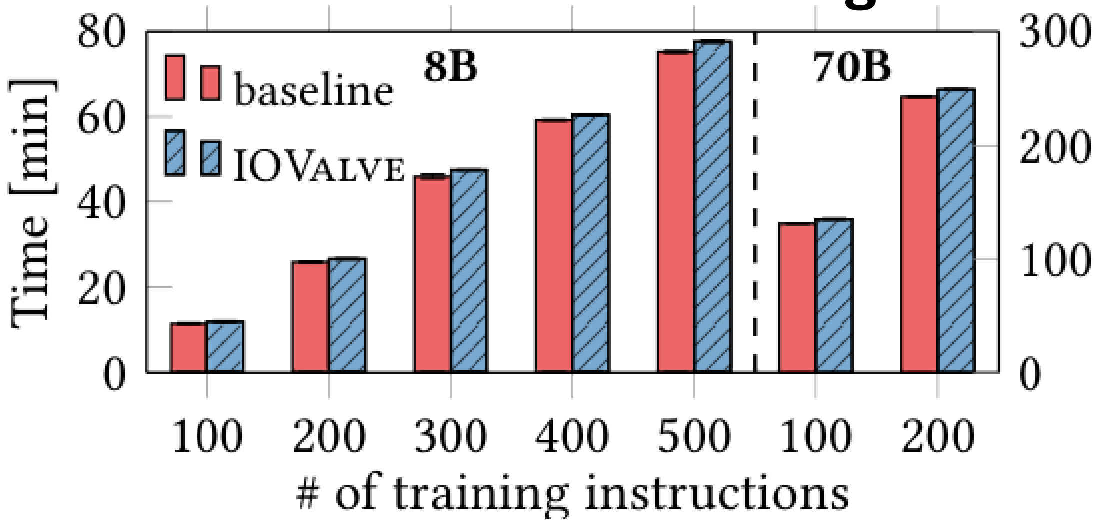
Evaluation: Collective communication



- Overhead is inversely proportional to message size.
  - Small message: 60.9%-6.1x
  - Large message: 18.7%-39.1%
- For small ones, our fixed-size messages mostly filled with dummies.

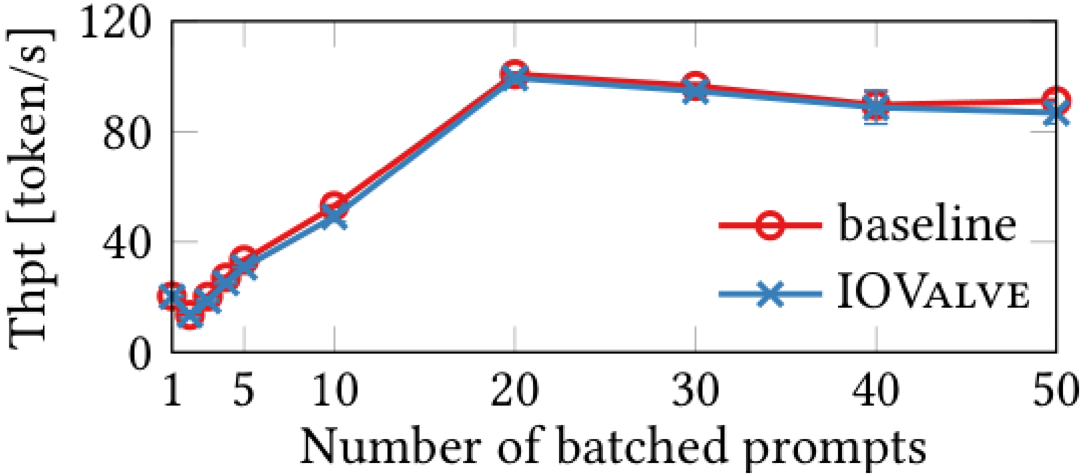
Evaluation: Real-world applications

Llama 3 fine tuning



Overhead: 2.1%-4.1%

Llama 3 (8B) batch inference



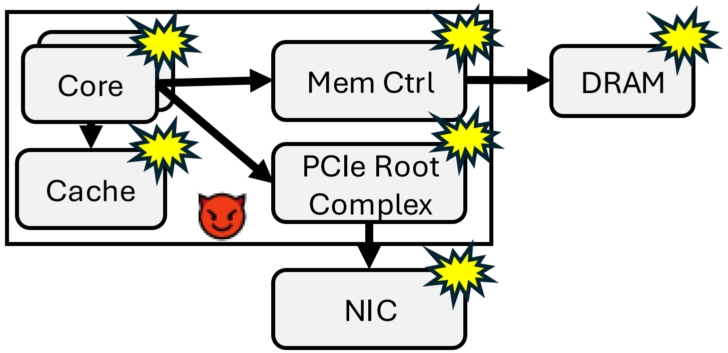
Overhead: 0.2%-8.5%

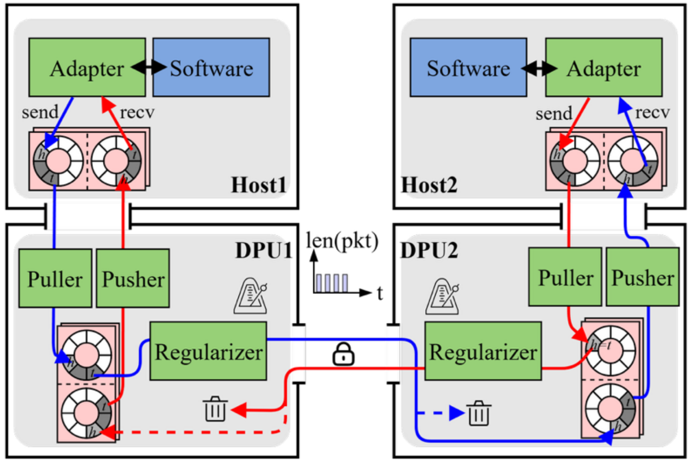
Takeaway: Let's rethink sandbox boundary

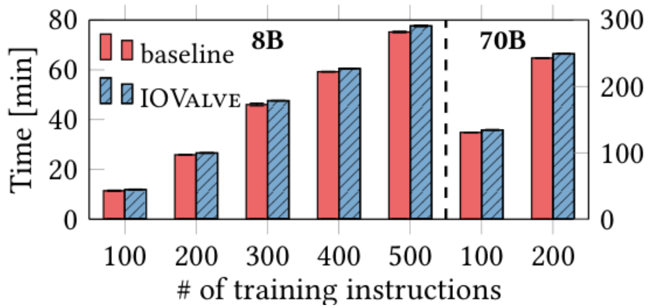
Existing sandbox overlooks side & covert channels

I/O sandbox with physical isolation

Run real-world workloads with low overhead







Contact: Sangho.Lee@microsoft.com