SecLab
COMPUTER SECURITY LABORATORY

15th Oct., 2025

GPU Travelling

Efficient Confidential Collaborative Training with TEE-Enabled GPUs

Shixuan Zhao[†], Zhongshu Gu[‡], Salman Ahmed[‡], Enriquillo Valdez[‡], Hani Jamjoom[‡], Zhiqiang Lin[†]

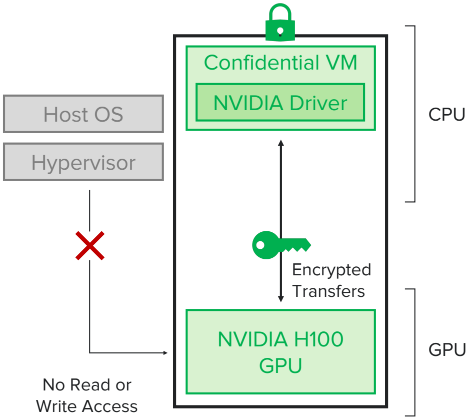
[†]The Ohio State University
[‡]IBM Research

IBM
THE OHIO STATE UNIVERSITY
COLLEGE OF ENGINEERING

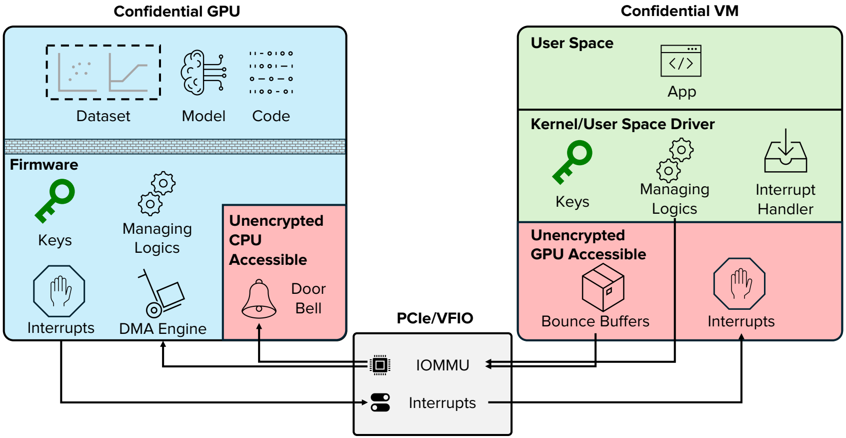
GPU TEEs

NVIDIA Confidential Computing

- Support added from Hopper (H100)
- No trust to the hypervisor
- Encrypt PCIe communication via driver and firmware



GPU TEEs


Confidential GPU
Dataset
Model
Code
Firmware
Keys
Managing Logics
Unencrypted CPU Accessible
Door Bell
Interrupts
DMA Engine
PCIe/VFIO
IOMMU
Interrupts
Confidential VM
User Space
App
Kernel/User Space Driver
Keys
Managing Logics
Interrupt Handler
Unencrypted GPU Accessible
Bounce Buffers
Interrupts

Confidential collaborative training

What is

- Multiple data owners
- Mutually distrusted
- One model

Medical Industry

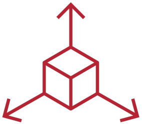
Media Industry

Existing solutions

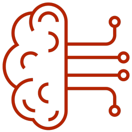
Existing Solutions

- Share the datasets
- Share the model/gradients

Dataset



vs.

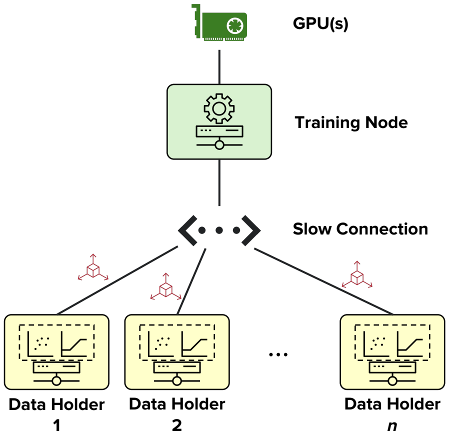


Model

Existing solutions

Sharing dataset

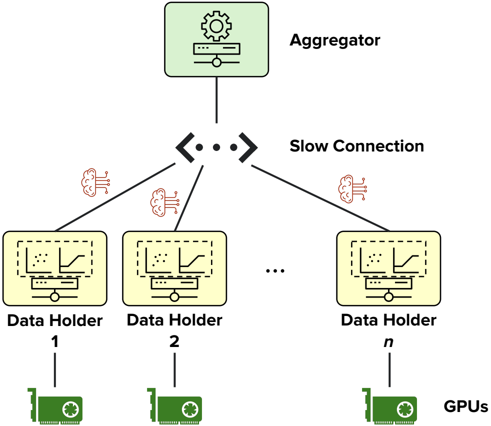
- Centralised training
- ✓ No model consolidation needed
- × Large datasets go through slow connections
- × Sensitive datasets travel a long data path



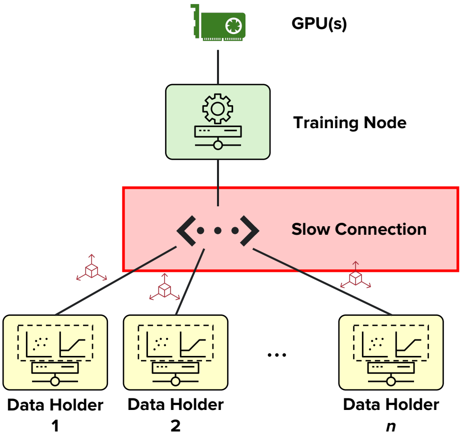
Existing solutions

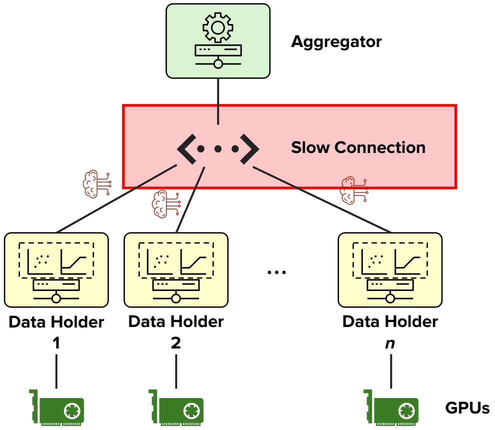
Sharing model/gradients

- E.g., federated learning
- ✓ Datasets don't travel a long data path
- × Large model/gradients go through slow connections
- × Extra overheads on model consolidation
- × Everyone needs enough GPUs

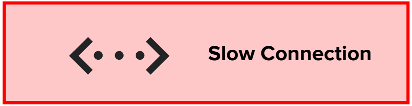


The common problem





The common problem

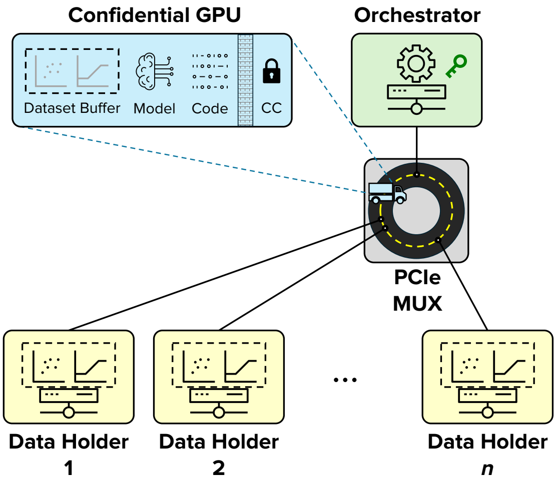


Can we just eliminate this?

Our solution

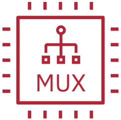
GPU Travelling

- An orchestrator
- Multiple Data Holders
- A Travelling GPU



Our solution

Physical Layer Travelling



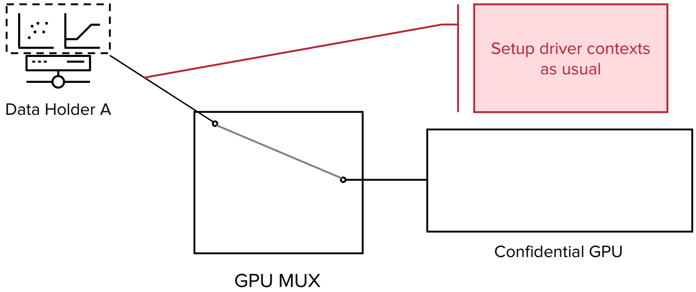
PCIe MUX: Routing a physical GPU to multiple VMs

Encrypted transfer via PCIe:
Sharing the key = sharing the GPU

Security Layer Travelling

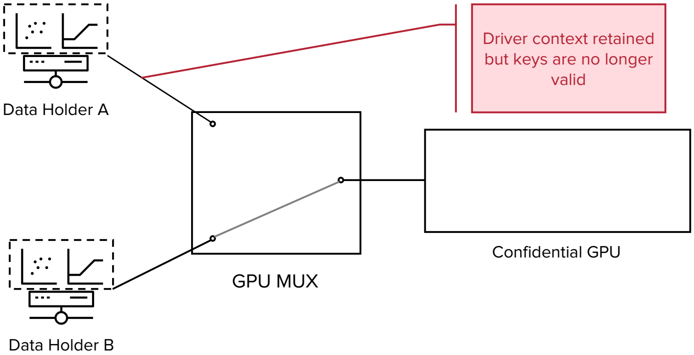
How it works - Initialisation

Booting



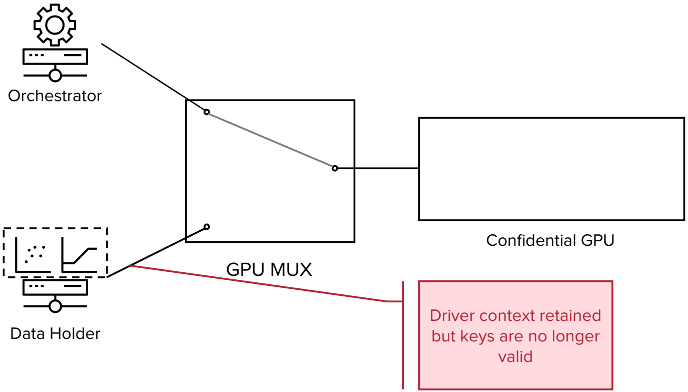
How it works - Initialisation

Booting



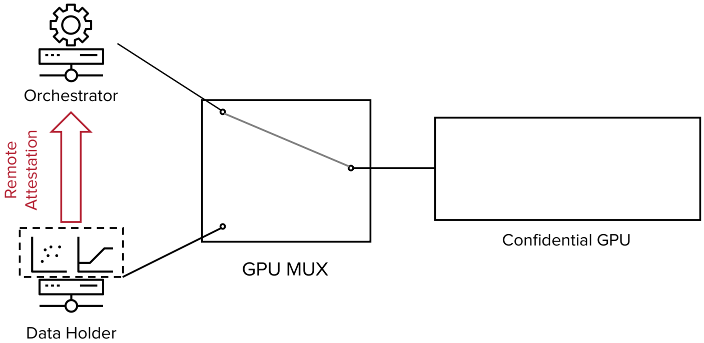
How it works - Initialisation

Booting



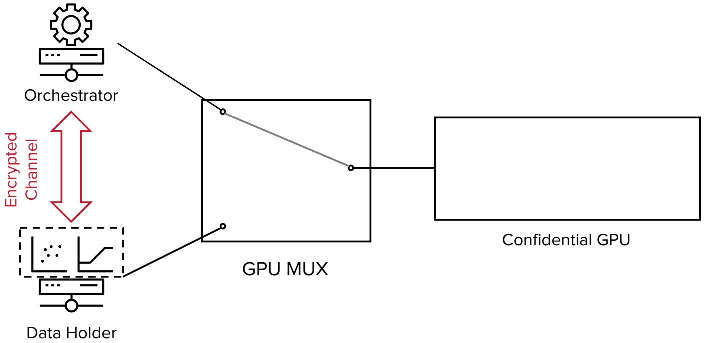
How it works - Initialisation

Booting



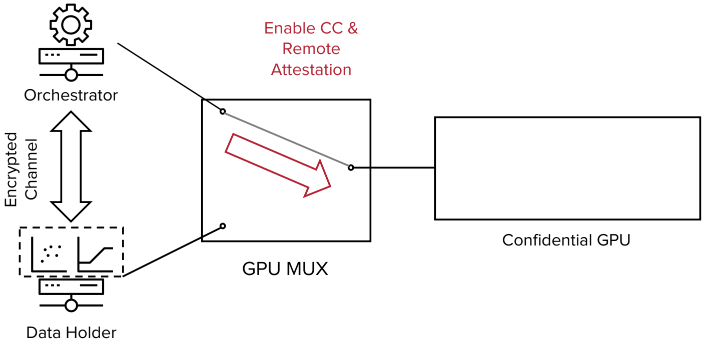
How it works - Initialisation

Booting



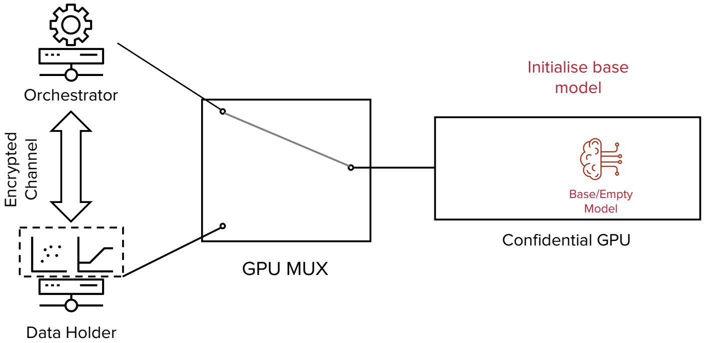
How it works - Initialisation

Setup



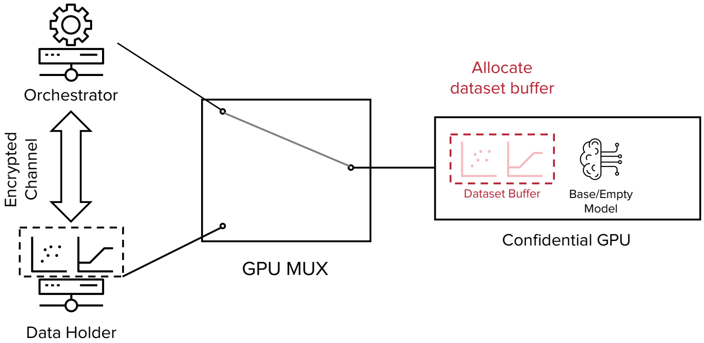
How it works - Initialisation

Setup



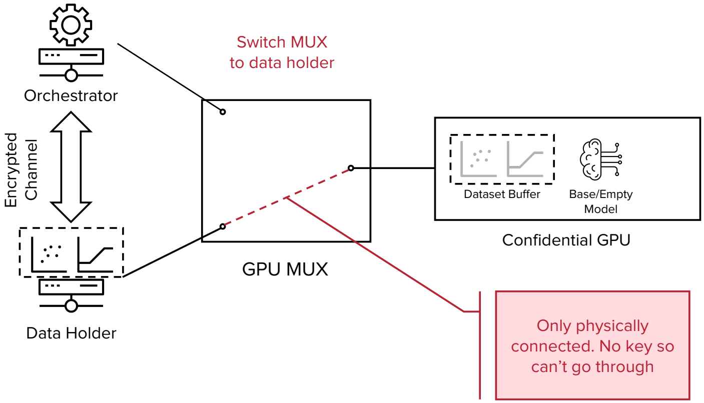
How it works - Initialisation

Setup



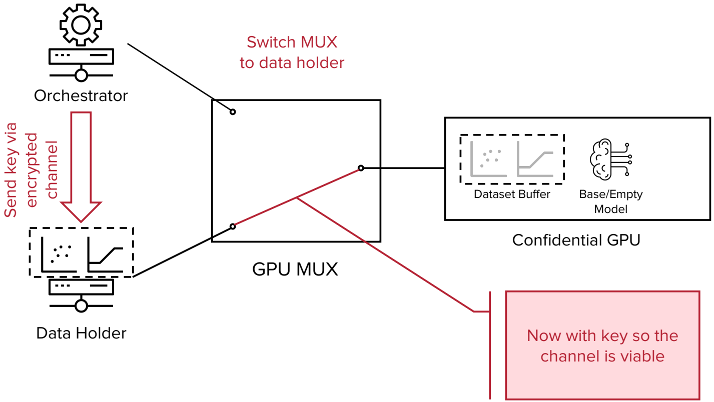
How it works - Travelling

Physical layer travelling



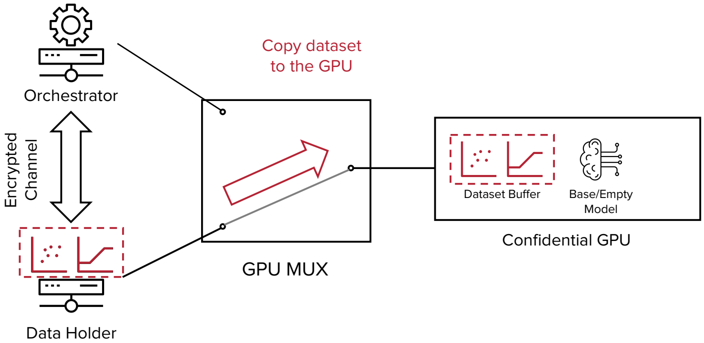
How it works - Travelling

Security layer travelling



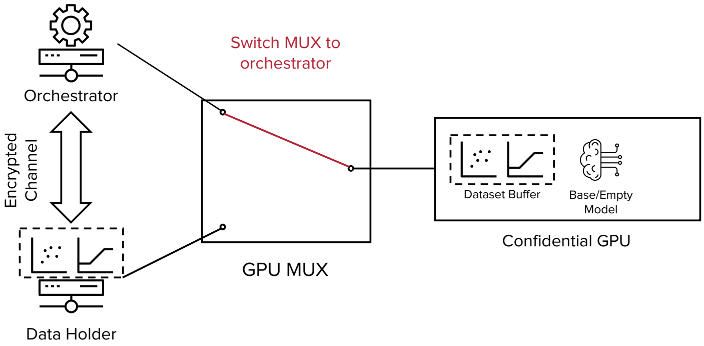
How it works - Travelling

Data provisioning



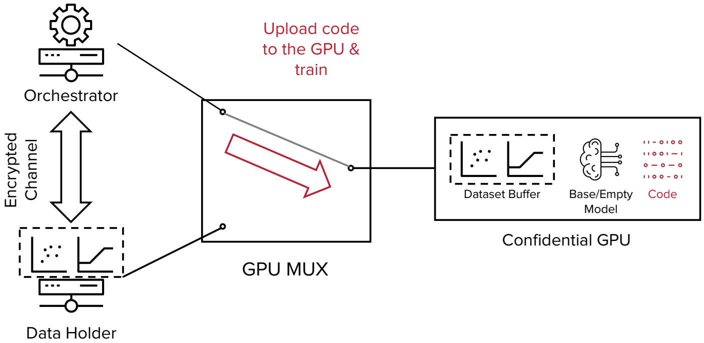
How it works - Travelling

Switch back



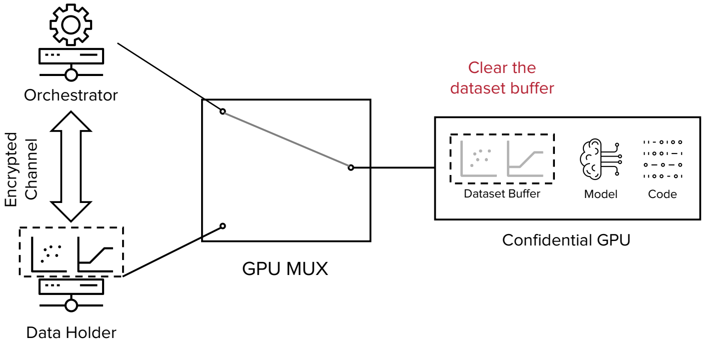
How it works - Travelling

Training



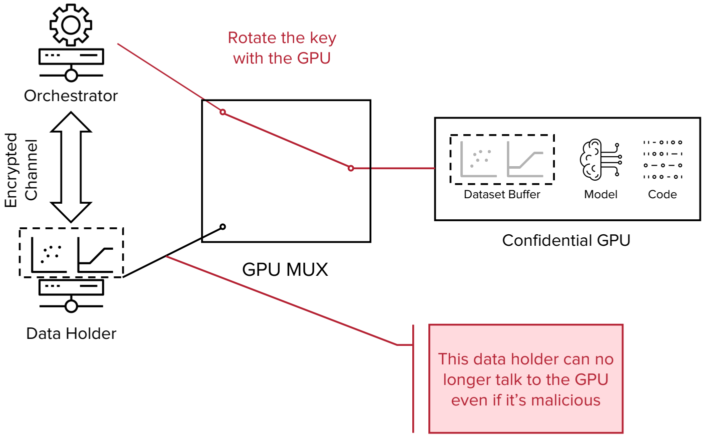
How it works - Epilogue

Dataset buffer clean up



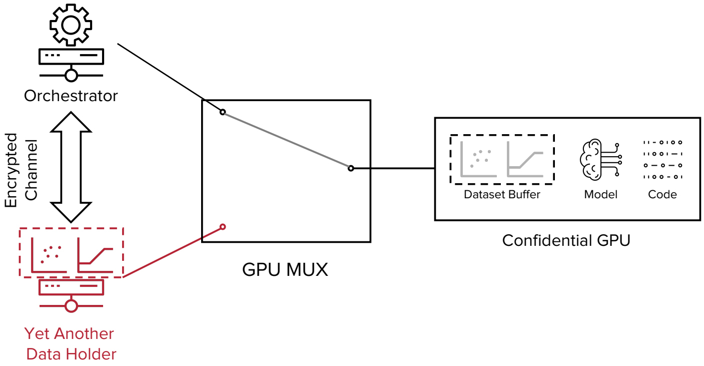
How it works - Epilogue

Key rotation



How it works - Epilogue

Ready for the next one



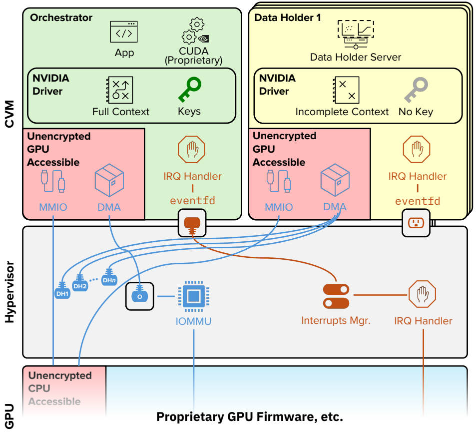
Ready to be passed to another data holder

Implementation

Requirement

No change to proprietary stuff

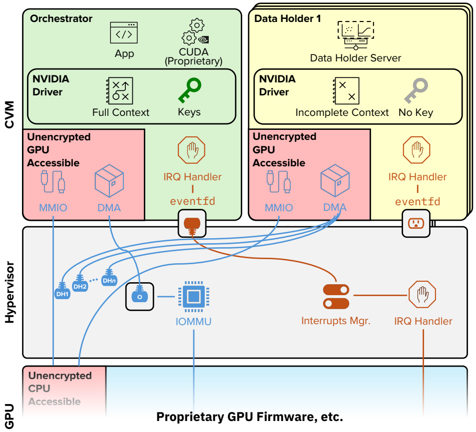
- Intel TDX
- NVIDIA H100
- VFIO-based MUX
- Modified NVIDIA driver
  - Key import/export
  - Context sync
  - Other ~~magic~~ chores



Implementation

- Total code changes: 4,746 LoC
- Artefacts available

https://zenodo.org/records/16899384



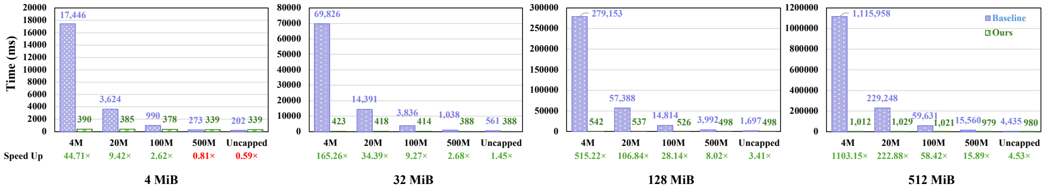
Evaluation

We've tried the real deal

- llm.c-based demo
- Showed significant efficiency improvement

- Demo artefacts also included

Data transfer overheads



The bigger the dataset buffer, the faster we are

llm.c comparison w/ GPT-2

- Save 7 seconds per 256 MiB transmission

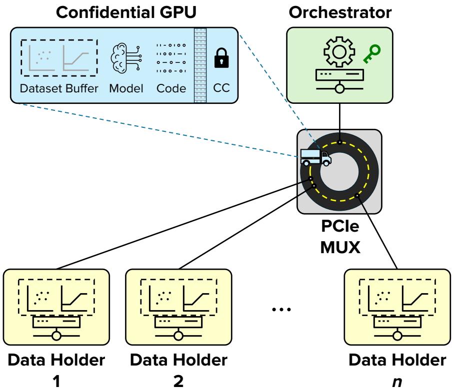
- Fineweb is 44 TiB in size
- 1261568 s (14+ days) for the entire dataset

	Baseline			Ours		
	Training (s)	Tx (s)	Tx Percentage	Training (s)	Tx (s)	Tx Percentage
4M	1230.00	1115.88	47.568%	1230.26	1.01	0.082%
20M	1231.39	230.84	15.787%	1229.39	1.03	0.084%
100M	1231.17	60.36	4.674%	1230.57	1.02	0.083%
500M	1230.73	15.86	1.272%	1229.67	0.98	0.080%
Uncapped	1229.36	7.34	0.594%	1231.46	0.98	0.079%

Conclusion

GPU Travelling significantly improved performance of confidential collaborative training

Eliminated transmission over slower channel by letting GPU to travel to the data holder to collect datasets directly



Thank you

Artefacts

https://zenodo.org/records/16899384

OSU SecLab

https://go.osu.edu/seclab

IBM Research

https://research.ibm.com