USING GLOBUS COMPUTE TO STREAMLINE FEDERATED LEARNING APPLICATIONS

ZILINGHAN LI

Machine Learning Engineer
Data Science and Learning Division,
Argonne National Laboratory
zilinghan.li@anl.gov

RAVI MADDURI

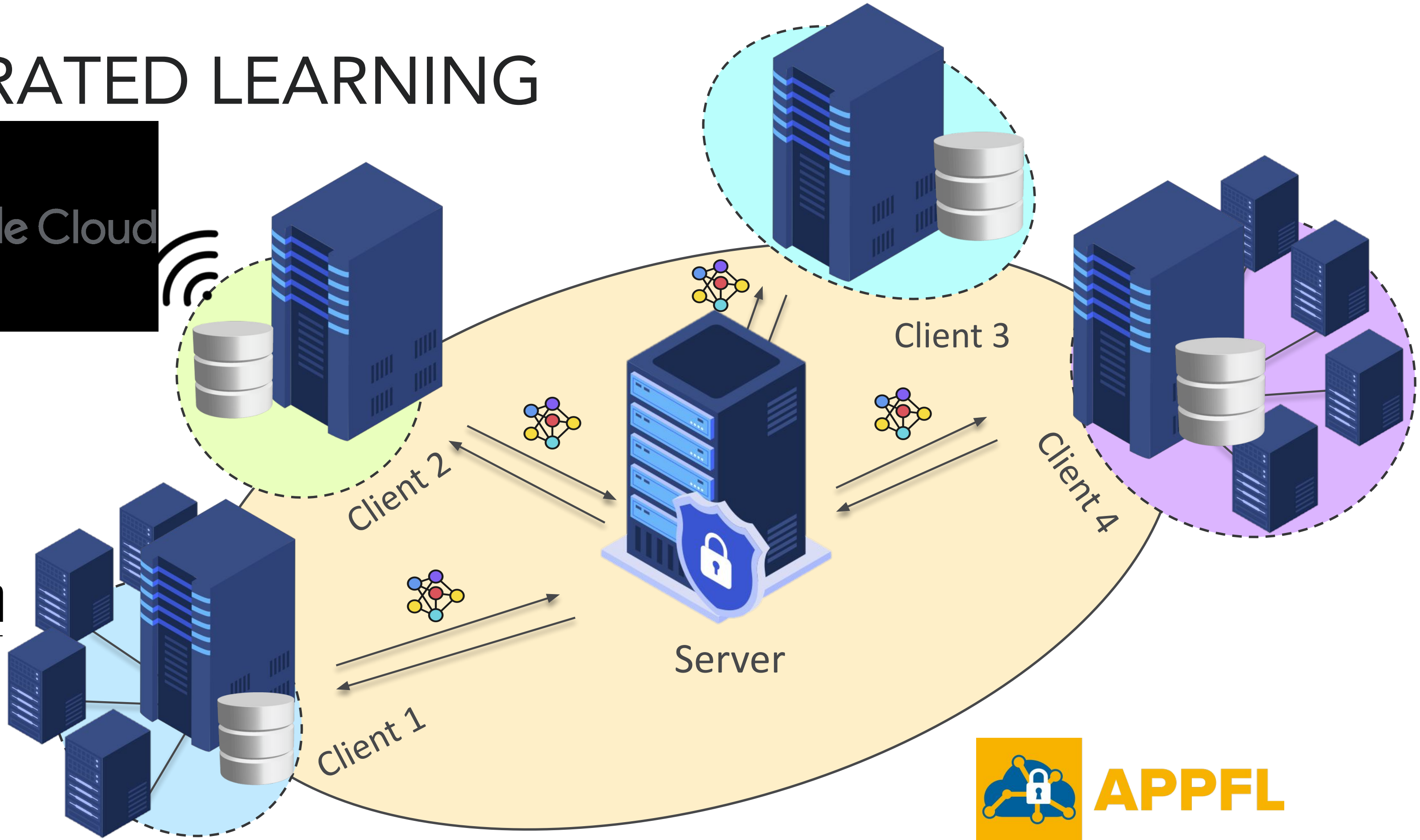
Senior Computer Scientist
Data Science and Learning Division,
Argonne National Laboratory
madduri@anl.gov

ParslFest 2024

FEDERATED LEARNING

Google Cloud

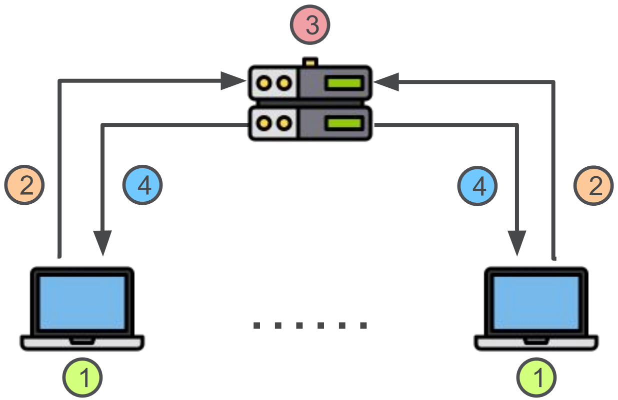
slurm
workload manager



APPFL

COMMUNICATION IN FEDERATED LEARNING

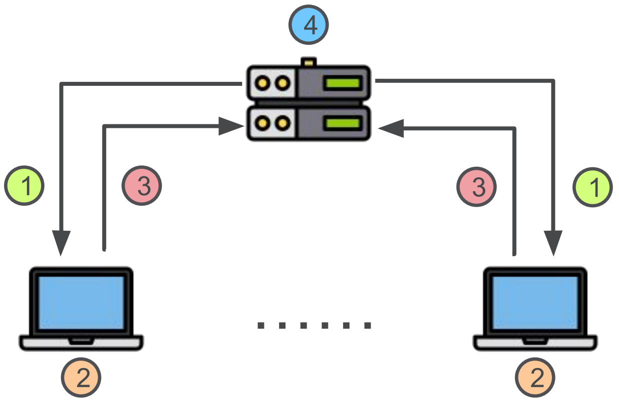
Client Driven and Server Driven Communications



1. Perform local training
2. Request global aggregation
3. Perform global aggregation
4. Send aggregated model

(a) Client-driven communication

gRPC



1. Send local training task
2. Perform local training
3. Send locally trained model
4. Perform global aggregation

(b) Server-driven communication

globus

BENEFITS OF GLOBUS COMPUTE

What benefits does Globus Compute (Server-driven communication) provide?

✅ Simple Experiment Launching and Testing

✅ Simple Experiment Coordination

All codes and configurations reside on the server side, making experiment launching, code/configuration updating, etc. as easy as serial experiments – there is no need to update code for each client one by one

✅ Robust Identity and Access Management

Simplifies the process to coordinate distributed training on heterogeneous computing resources (e.g., with different job schedulers) – there is no need for each client to start "client launching job" nearly at the same time.

Globus Compute integrates with Globus authentication for robust access management.

✅ No Inbound Connectivity Requirements

Both the FL server and FL clients only require outbound traffic, without any inbound traffic requirements, making resources FL server be setup on resources like Polaris.

FL ON HETEROGENEOUS CLIENTS

Globus Compute Enables FL on Heterogeneous Clients

Google Cloud

aws

Heterogeneous client computing resources.

Resource under-utilization, especially for powerful client machines

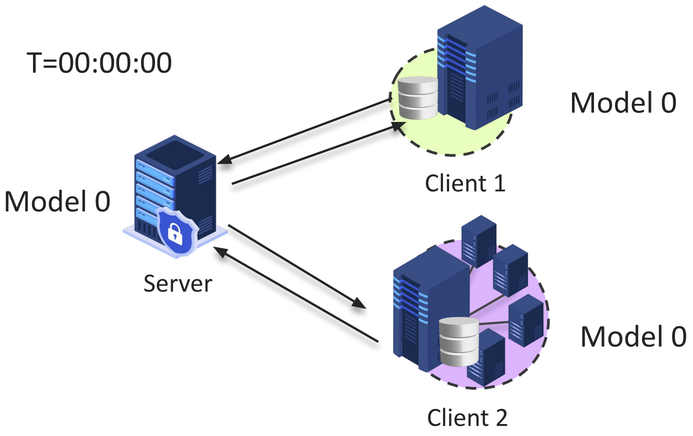
▶	client_training	crn-azure	2023-06-14 18:01:56	2023-06-14 18:02:12	15.66 sec
▶	client_training	Polaris	2023-06-14 18:01:56	2023-06-14 18:02:12	15.83 sec
▶	client_training	delta-cpu-01	2023-06-14 18:01:56	2023-06-14 18:02:28	31.97 sec
▶	client_training	delta-cpu02	2023-06-14 18:01:56	2023-06-14 18:02:35	39.09 sec

Different amount of local training times on heterogeneous client machines.

RESOLVING HETEROGENEOUS CLIENTS

Asynchronous Federated Learning

- Asynchronous FL updates global model immediately once receiving local model from each client – suffers from the stale (outdated) local models from slower clients, thereby causing the global model to drift away from slower clients.

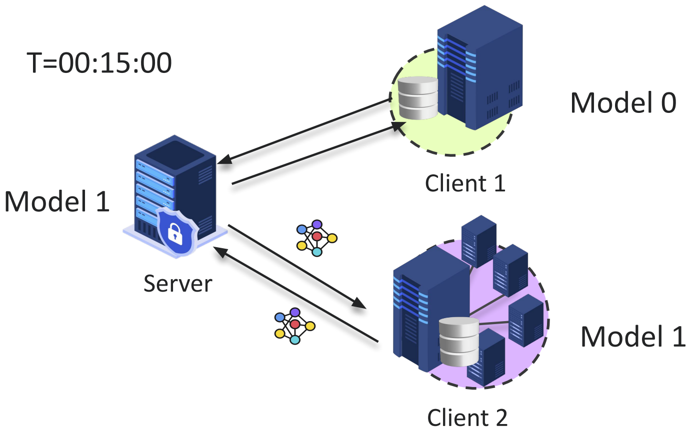


Staleness problem in asynchronous FL.

RESOLVING HETEROGENEOUS CLIENTS

Asynchronous Federated Learning

- Asynchronous FL updates global model immediately once receiving local model from each client – suffers from the stale (outdated) local models from slower clients, thereby causing the global model to drift away from slower clients.

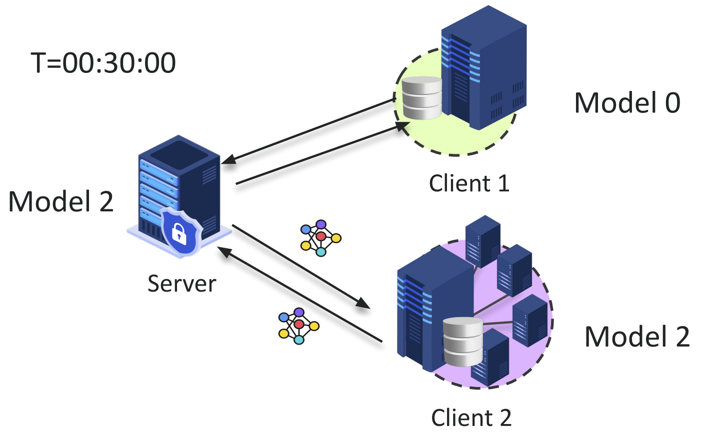


Staleness problem in asynchronous FL.

RESOLVING HETEROGENEOUS CLIENTS

Asynchronous Federated Learning

- Asynchronous FL updates global model immediately once receiving local model from each client – suffers from the stale (outdated) local models from slower clients, thereby causing the global model to drift away from slower clients.

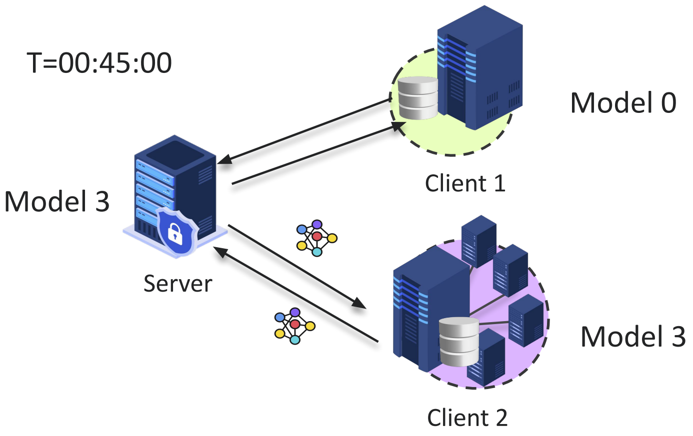


Staleness problem in asynchronous FL.

RESOLVING HETEROGENEOUS CLIENTS

Asynchronous Federated Learning

- Asynchronous FL updates global model immediately once receiving local model from each client – suffers from the stale (outdated) local models from slower clients, thereby causing the global model to drift away from slower clients.

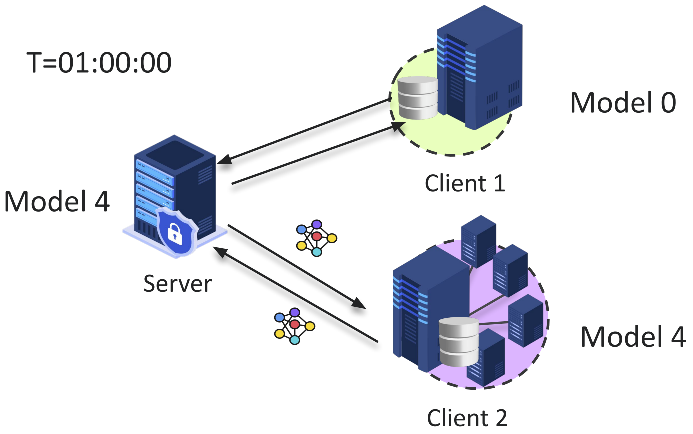


Staleness problem in asynchronous FL.

RESOLVING HETEROGENEOUS CLIENTS

Asynchronous Federated Learning

- Asynchronous FL updates global model immediately once receiving local model from each client – suffers from the stale (outdated) local models from slower clients, thereby causing the global model to drift away from slower clients.

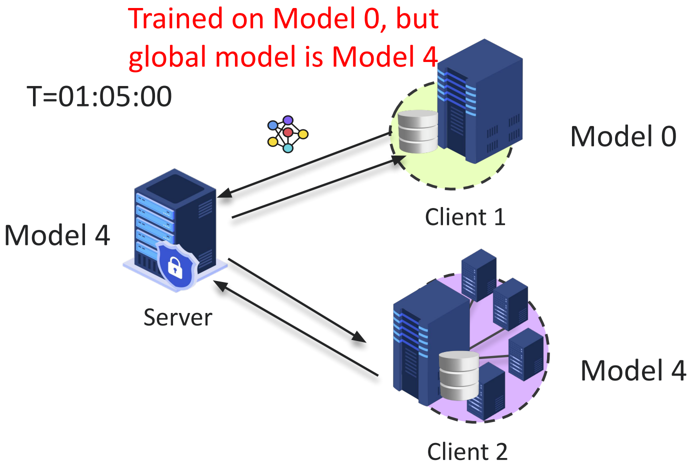


Staleness problem in asynchronous FL.

RESOLVING HETEROGENEOUS CLIENTS

Asynchronous Federated Learning

- Asynchronous FL updates global model immediately once receiving local model from each client – suffers from the stale (outdated) local models from slower clients, thereby causing the global model to drift away from slower clients.



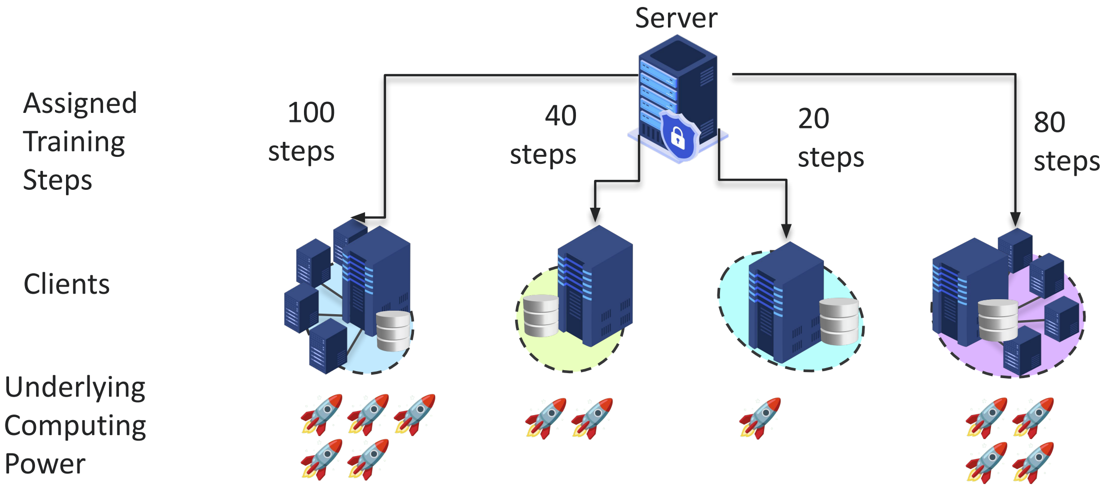
Staleness problem in asynchronous FL.

Local model from slower clients are stale/oudated compared to the global model!

(1) Either be detrimental to global model;
(2) or applying a small importance weight and causing client drift.

RESOLVING HETEROGENEOUS CLIENTS

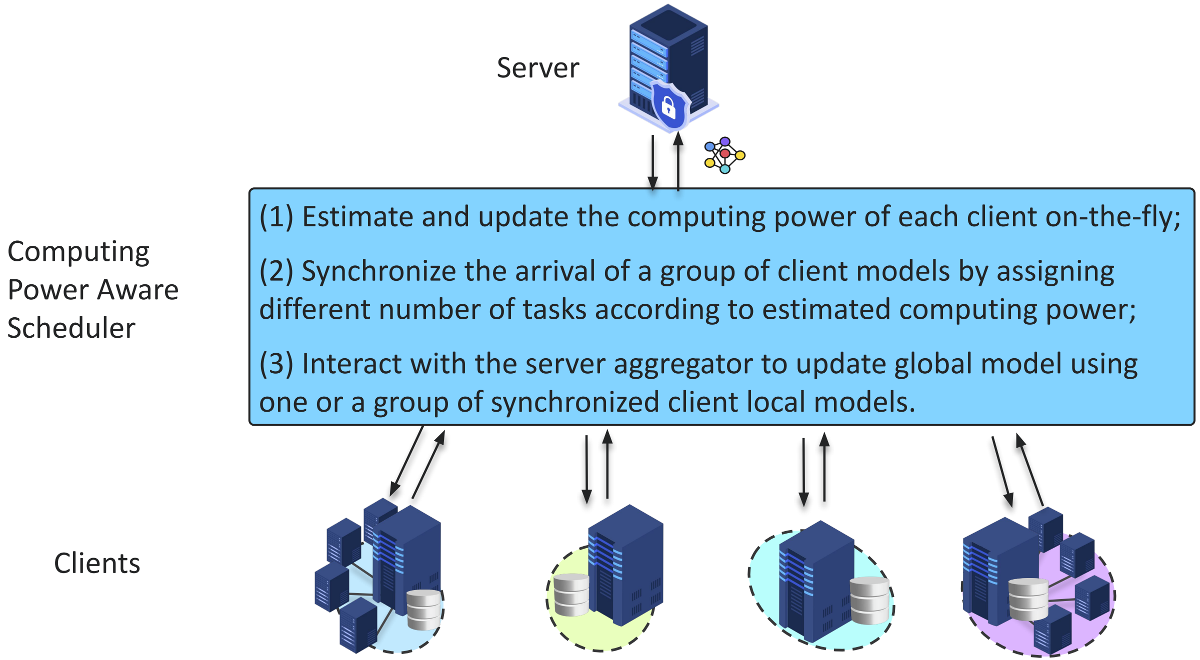
- "Synchronize" the arrival of clients' locally trained models
  - by assigning different numbers of local training steps to them
  - according to the clients' computing power



Assigning local training steps proportional to client's computing power.

However, in practice
(1) The server does not know the clients' computing power beforehand;
(2) And the computing power may change during the training.

RESOLVING HETEROGENEOUS CLIENTS

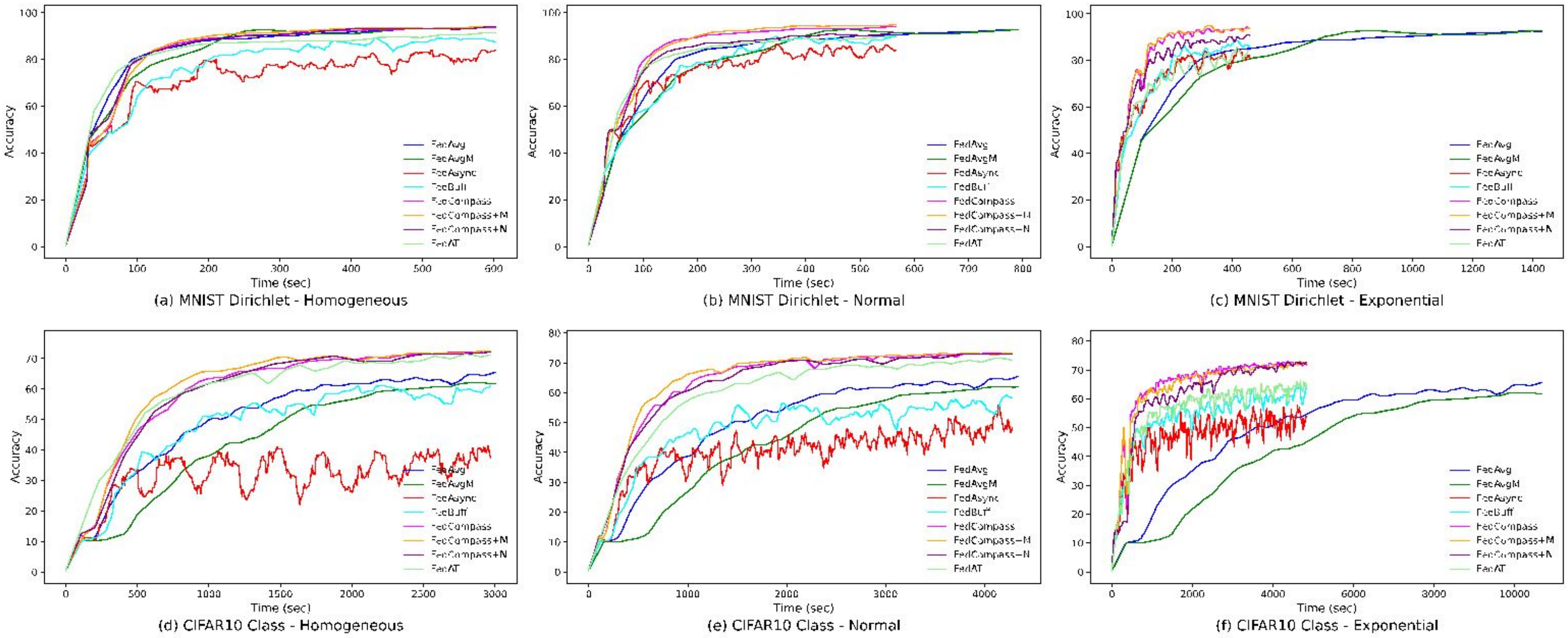


FedCompass - Federated learning with a computing power aware scheduler.

Li, Zilinghan, Pranshu Chaturvedi, Shilan He, Han Chen, Gagandeep Singh, Volodymyr Kindratenko, Eliu A. Huerta, Kibaek Kim, and Ravi Madduri. "FedCompass: efficient cross-silo federated learning on heterogeneous client devices using a computing power aware scheduler." *arXiv preprint arXiv:2309.14675* (2023).

RESOLVING HETEROGENEOUS CLIENTS

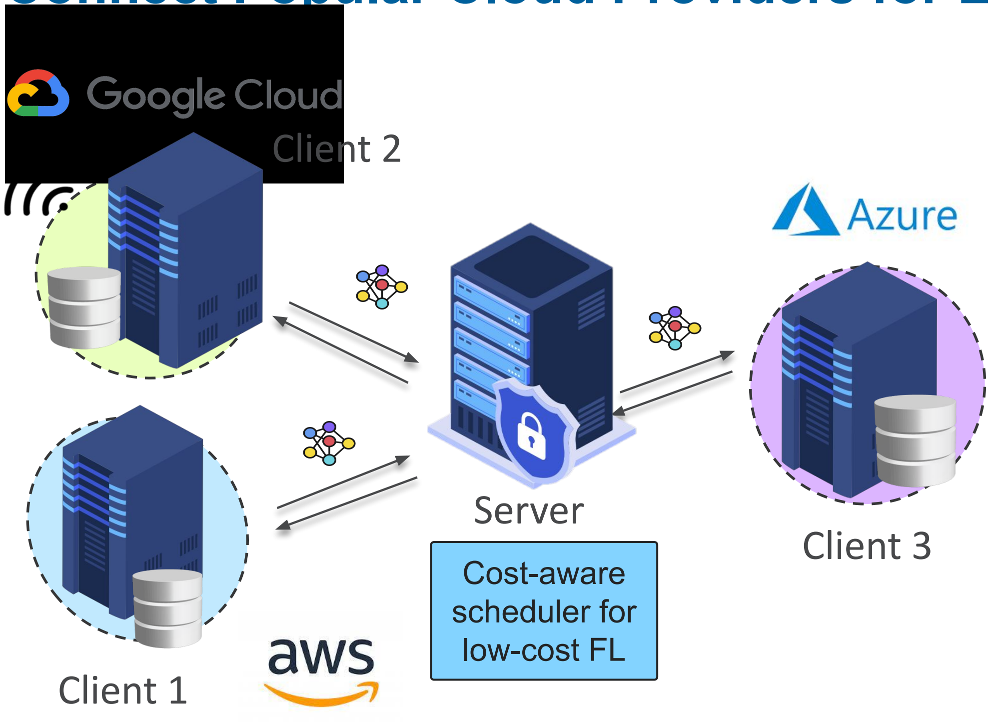
Results



Change in validation accuracy for various FL strategies during the training.

NEXT STEPS

Connect Popular Cloud Providers for Low-cost (Cost-aware) FL



- FL is important in medical applications, where data privacy is paramount.
- Many hospitals have their private data on Cloud Storage (S3, Globus Cloud Storage, etc.) and have their computing on the Cloud as well.
- Training on GPU cloud instances can be costly.
- AWS, Google, and Azure all have “spot computing” – AWS Spot Instances, Google Cloud Preemptable VMs, and Azure Spot VMs, which provide a low-cost computing option, but can be killed at any time with a short notice.
- We would like to add cost-aware aspects to compute-aware scheduler to **reduce the cost** for FL experiments among heterogeneous cloud computing providers using their spot instances, and make the setup process as **streamlined** as possible

REFERENCE

- https://github.com/APPFL/APPFL
- https://appfl.ai
- Li, Zilinghan, Pranshu Chaturvedi, Shilan He, Han Chen, Gagandeep Singh, Volodymyr Kindratenko, Eliu A. Huerta, Kibaek Kim, and Ravi Madduri. "FedCompass: Efficient Cross-Silo Federated Learning on Heterogeneous Client Devices Using a Computing Power-Aware Scheduler." In The Twelfth International Conference on Learning Representations.
- Li, Zilinghan, Shilan He, Ze Yang, Minseok Ryu, Kibaek Kim, and Ravi Madduri. "Advances in APPFL: A Comprehensive and Extensible Federated Learning Framework." arXiv preprint arXiv:2409.11585 (2024).

Q&A