TOWARDS TAMING THE RESOURCE AND DATA HETEROGENEITY IN FEDERATED LEARNING

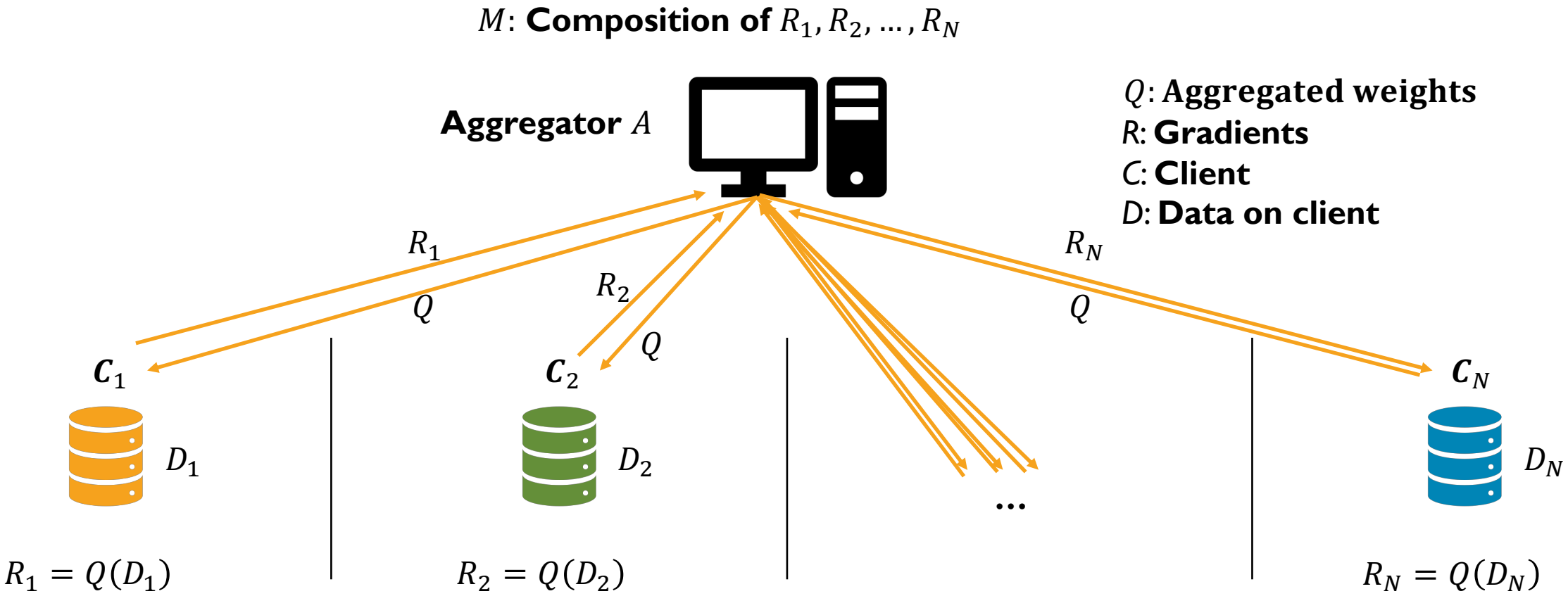
Zheng Chai[1], Hannan Fayyaz[2], Zeshan Fayyaz[3], Ali Anwar[4],
Yi Zhou4[4], Nathalie Baracaldo[4], Heiko Ludwig[4], Yue Cheng[1]

[1]*George Mason University,* [2]*York University,* [3]*Ryerson University,* [4]*IBM Research–Almaden*

Federated Learning - Background

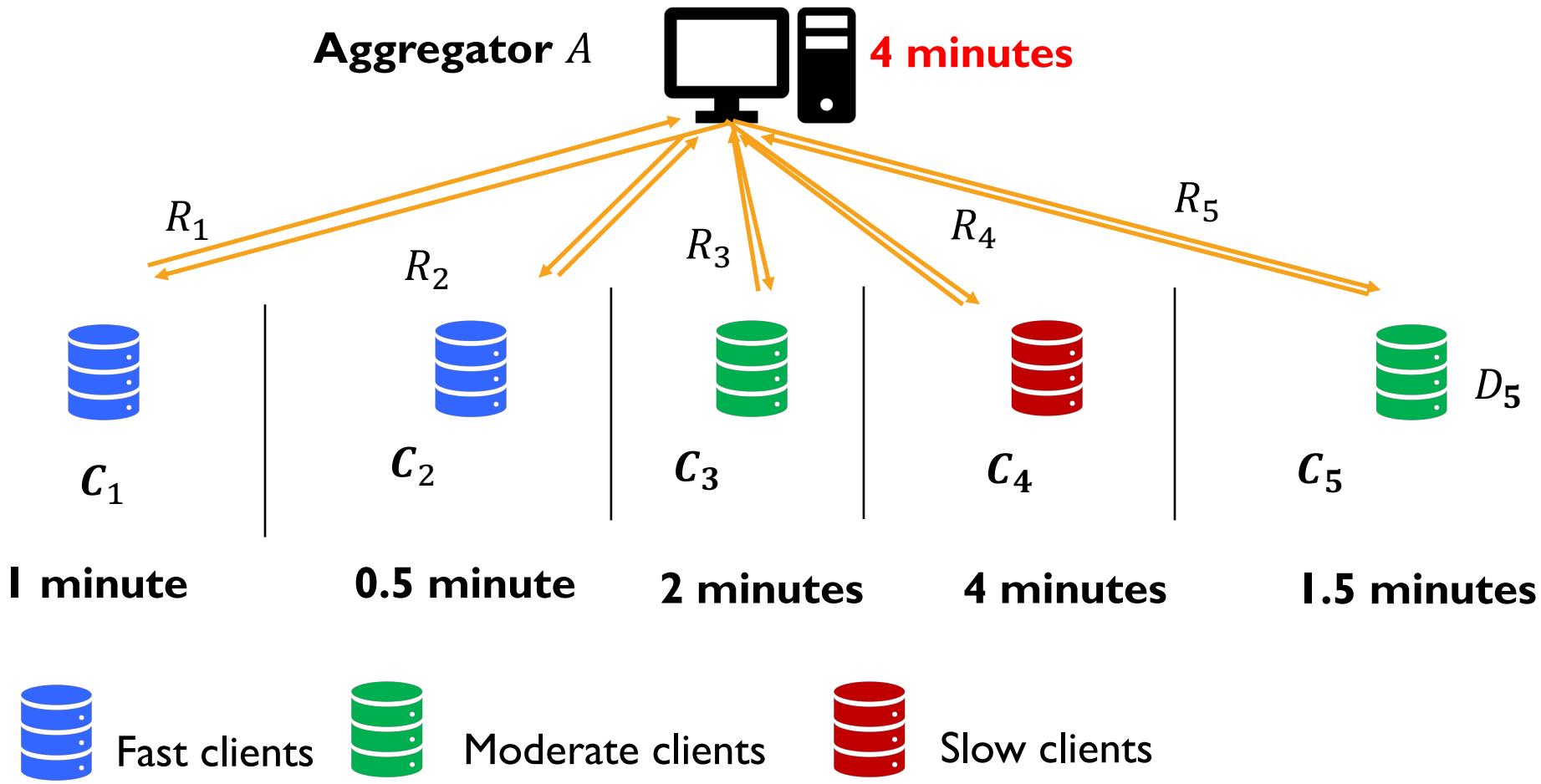
- **Federated learning - learning in distributed data environments**
  - In a federated learning system, multiple data sources collaborate to learn a predictive model
  - **Limited trust** is likely to exist between each client, so clients do not want other participants to learn their private data in the process
- **Limited communication – Data/Resource Heterogeneity**
  - **Data heterogeneity** some clients will have more data
  - **Resource heterogeneity** some clients may have lesser compute/connectivity
- **Clients can drop out during the training**

Federated Learning – Problem Statement

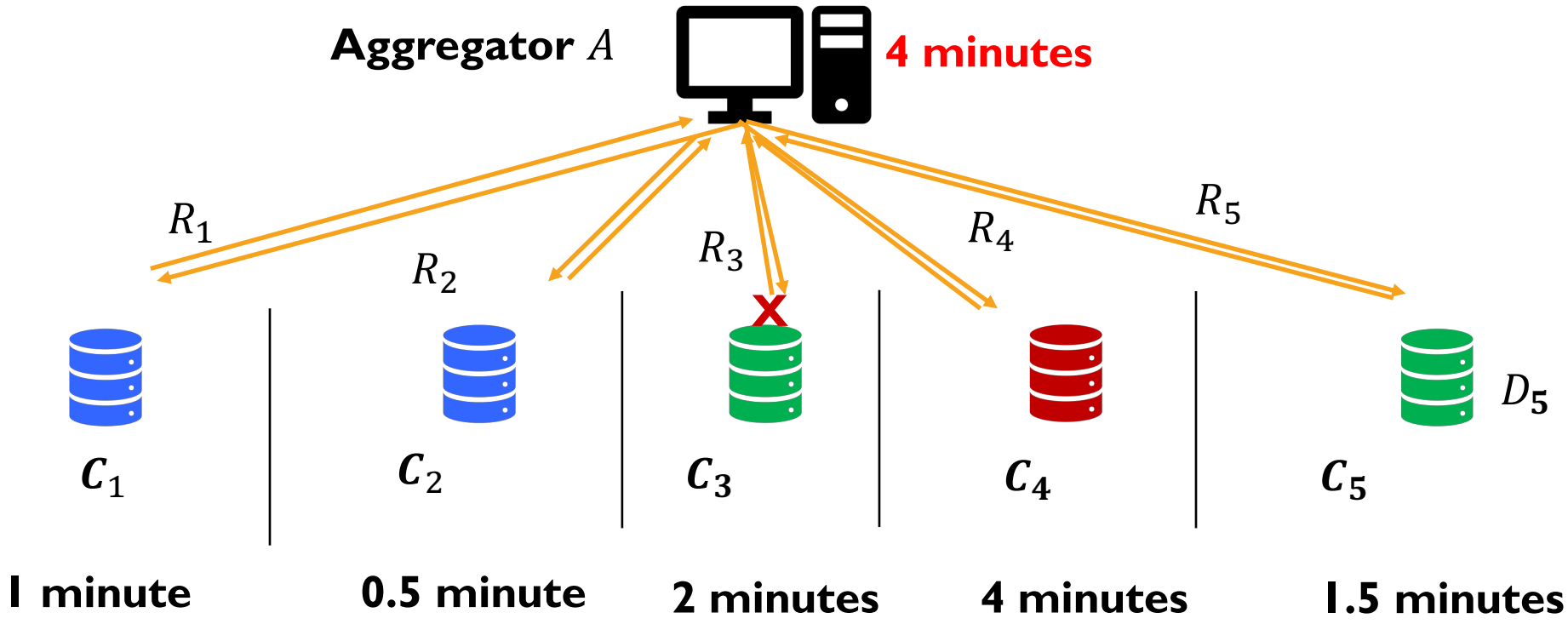


Issue: Aggregation happens after each client has replied

Federated Learning – Example Problem: Stragglers



Federated Learning – Example Problem: Dropouts



Methodology

Model: CNN (consists of two CNN layers and one Max Pooling layer)

Dataset: MNIST data

Non-IID: each client sample data from randomly selected 5 digit categories

No. of Total Clients: 100

Setup and Hyperparameters

Platform: AWS EC2 (m4.10xlarge with 40 vCPUs and 160 GB memory)

Library: TensorFlow

Hyperparameters:

Optimizer: Adadelta

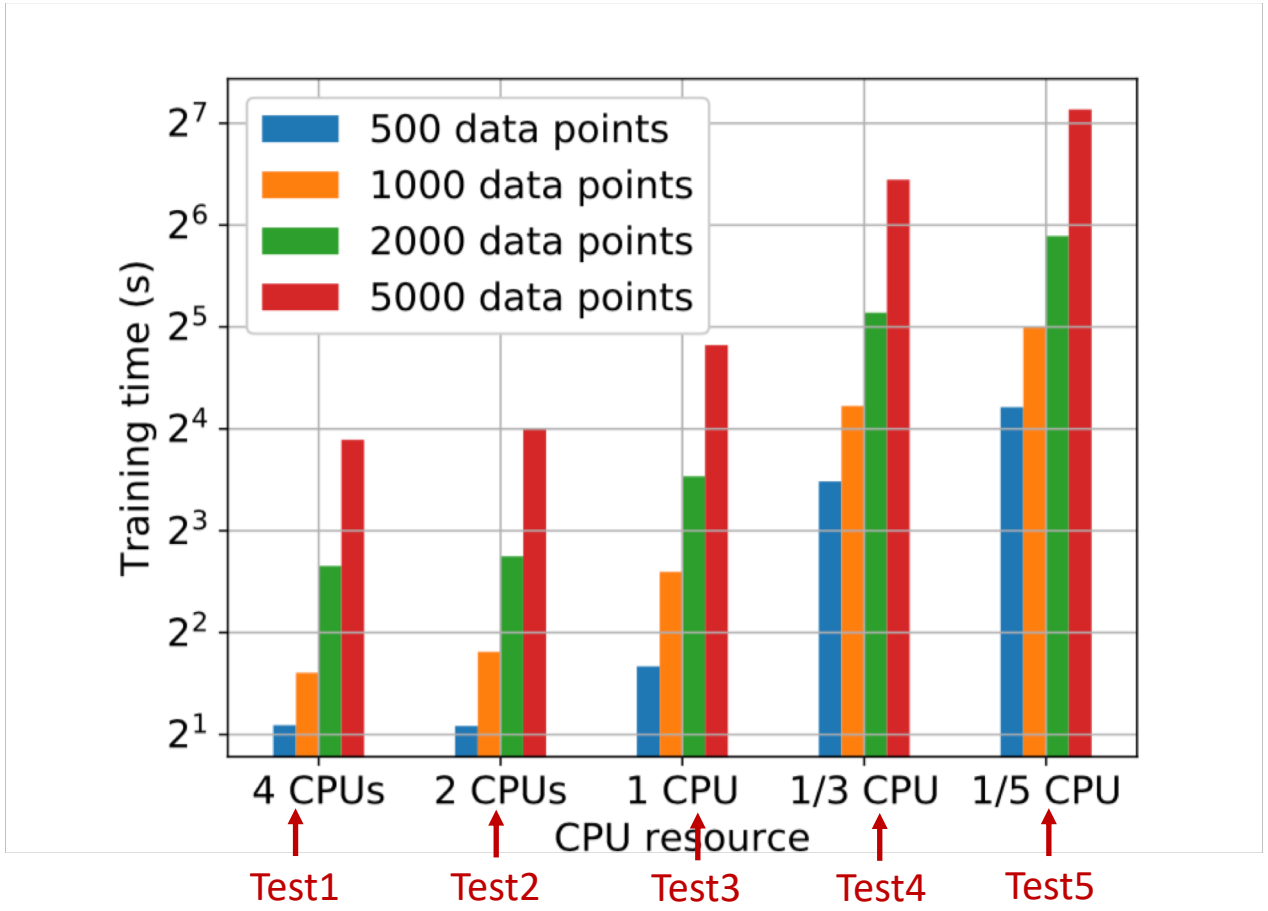
Epochs: 8 (each client per round)

Drop out rate: 0.25 & 0.5 for each CNN layer

Impact of resource heterogeneity on training time

Test	# of Clients	# of CPUs	CPUs per Client
1	4	16	4
2	4	8	2
3	4	4	1
4	3	1	1/3
5	5	1	1/5

Table 1: Distribution of data parties and CPUs.



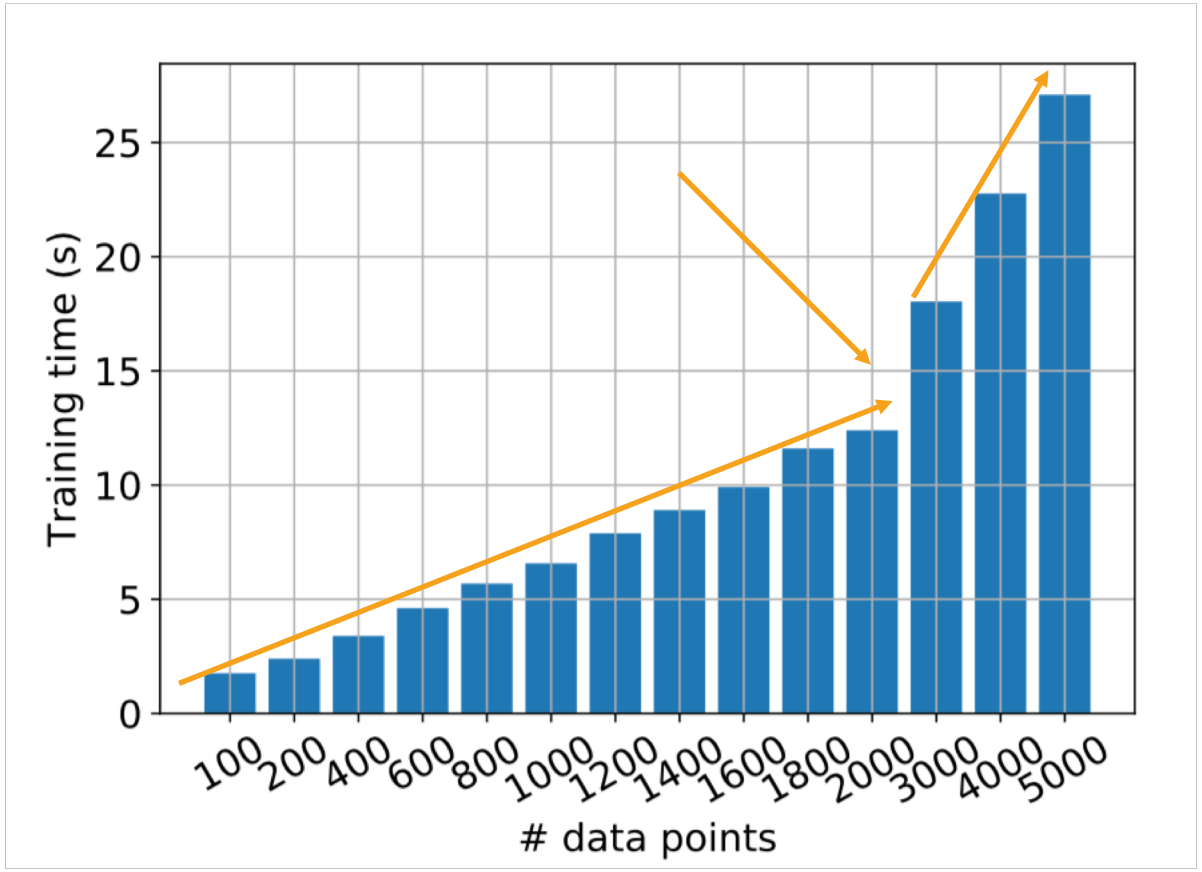
Per-round training time different CPU resources and different dataset sizes

Impact of data heterogeneity on training time

14 clients with different dataset size (varying from 100–5000 data points) but with the same amount of CPU resources

Training time gets linearly increased as the dataset size gets bigger

Data heterogeneity can significantly impact the FL system's training time



Per-epoch training time with different dataset sizes

Current ongoing work

Q1 *How to classify devices based on their response time and then use this information for our advantage without affecting the FL process?*

Q2 *How to incorporate data of each device in the FL process without worrying about stragglers?*

Q3 *How to identify drop-out devices and mitigate the effect of drop-out without affecting the ML process?*

Thanks!