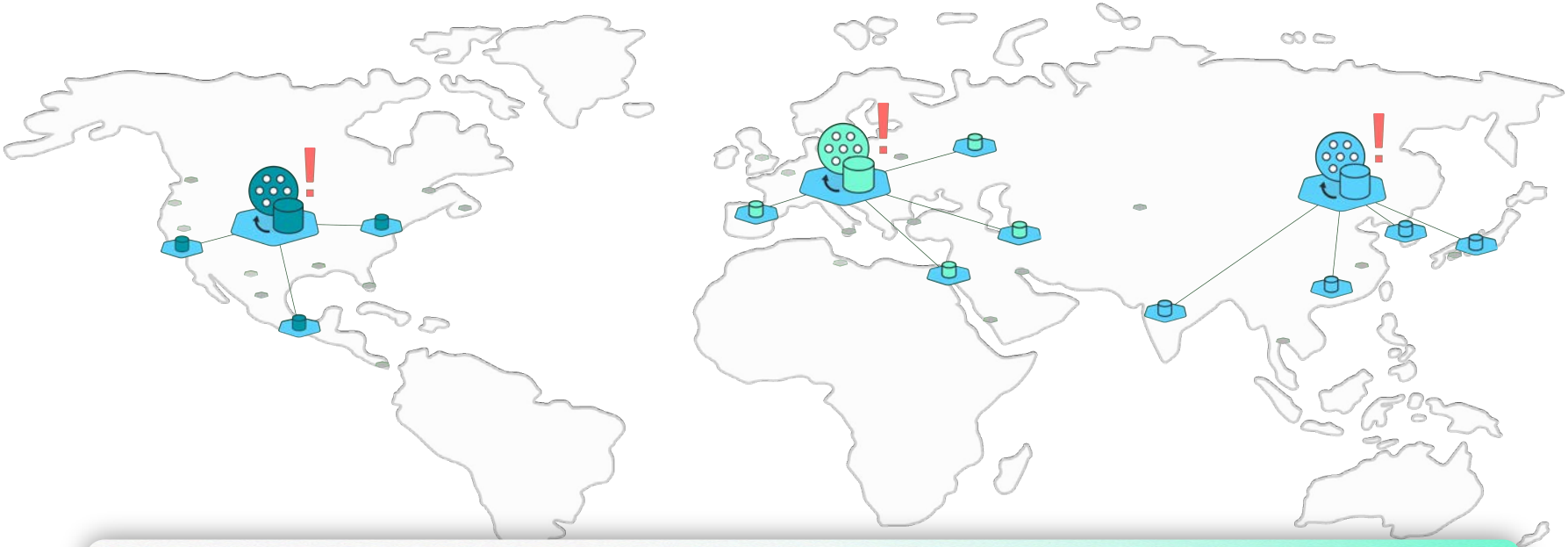
Flower

On-Device Federated Learning

On-Device Intelligence Workshop, MLSys 2021

Akhil Mathur, Daniel J. Beutel, Pedro Porto Buarque de Gusmão, Javier Fernandez-Marques, Taner Topal, Xinchi Qiu, Titouan Parcollet, Yan Gao, Nicholas D. Lane

Motivation



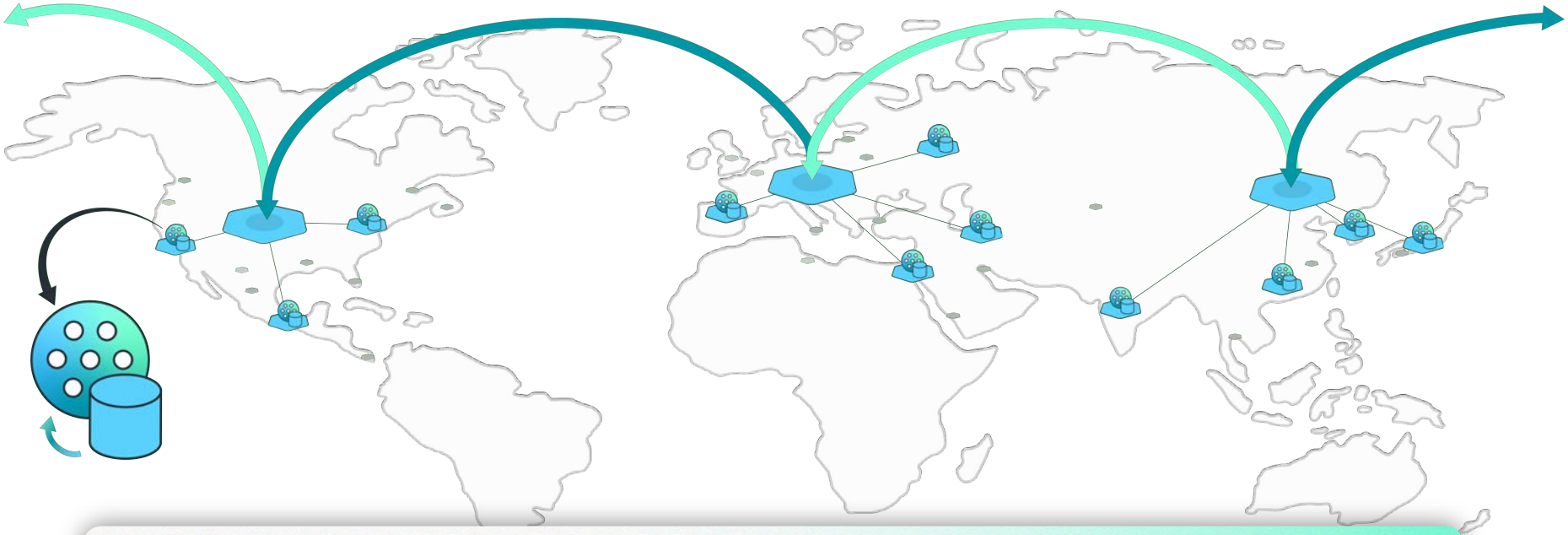
ML Today

Move data to model

- Battery
- Network
- Privacy/regulations

15

connected devices per person by 2030 (vs ~7 today)



Federated Learning

Move model to data

- Availability
- Bandwidth
- Regulations

+2.6b

new AI-enabled edge devices, yearly

Centralized Learning

Network:	**LAN**
Platform:	**Linux**
Hardware:	**CPU, GPU, TPU**
Framework:	**TensorFlow, PyTorch, JAX, MXNet,**
Protocol:	**gRPC**
Locality:	**single-region**
Data:	**IID**

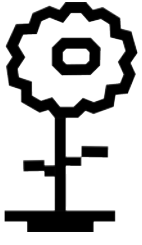
Federated Learning: Heterogeneity

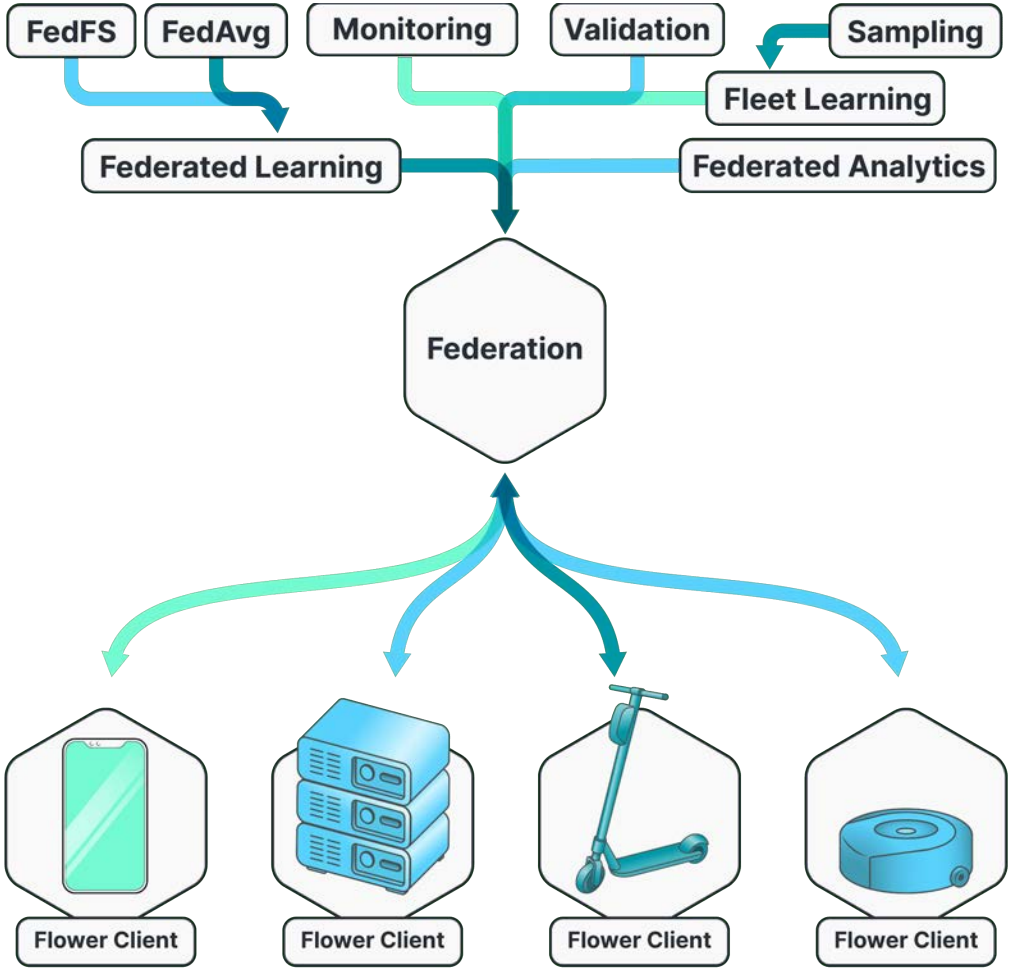
Network:	**LAN, WAN, WiFi, LoRaWan, 2/3/4/5/6G, BT, BT-LE, ...**
Platform:	**Linux, macOS, Windows, iOS, Android, embedded**
Hardware:	**CPU, GPU, TPU, edge-TPU, Neural Engine, ...**
Framework:	**TensorFlow, PyTorch, JAX, MXNet, libtorch, TF Lite, ...**
Protocol:	**gRPC, REST, MQTT, sockets, WebSockets, ...**
Locality:	**single-region, multi-region, global**
Data:	**IID, non-IID**

Flower

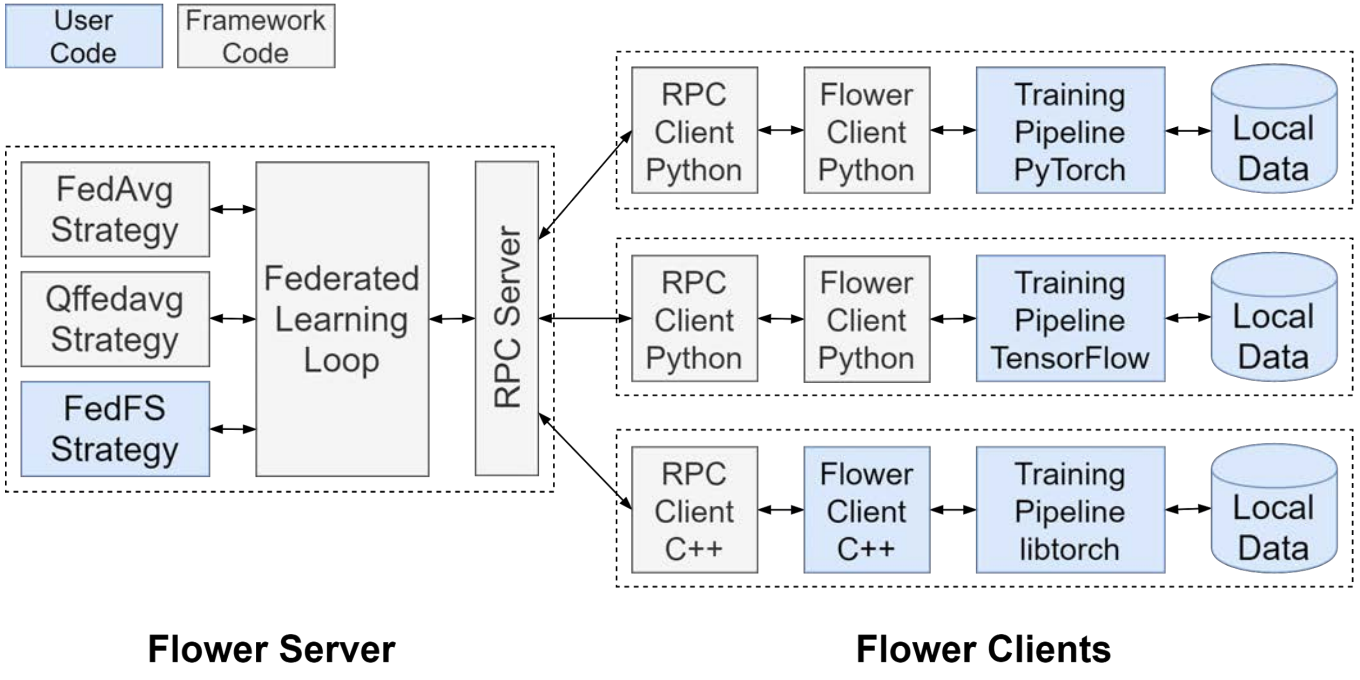
Flower
A Friendly Federation Framework

The Flower open source framework solves this complexity with modular components to accelerate the research of federated approaches

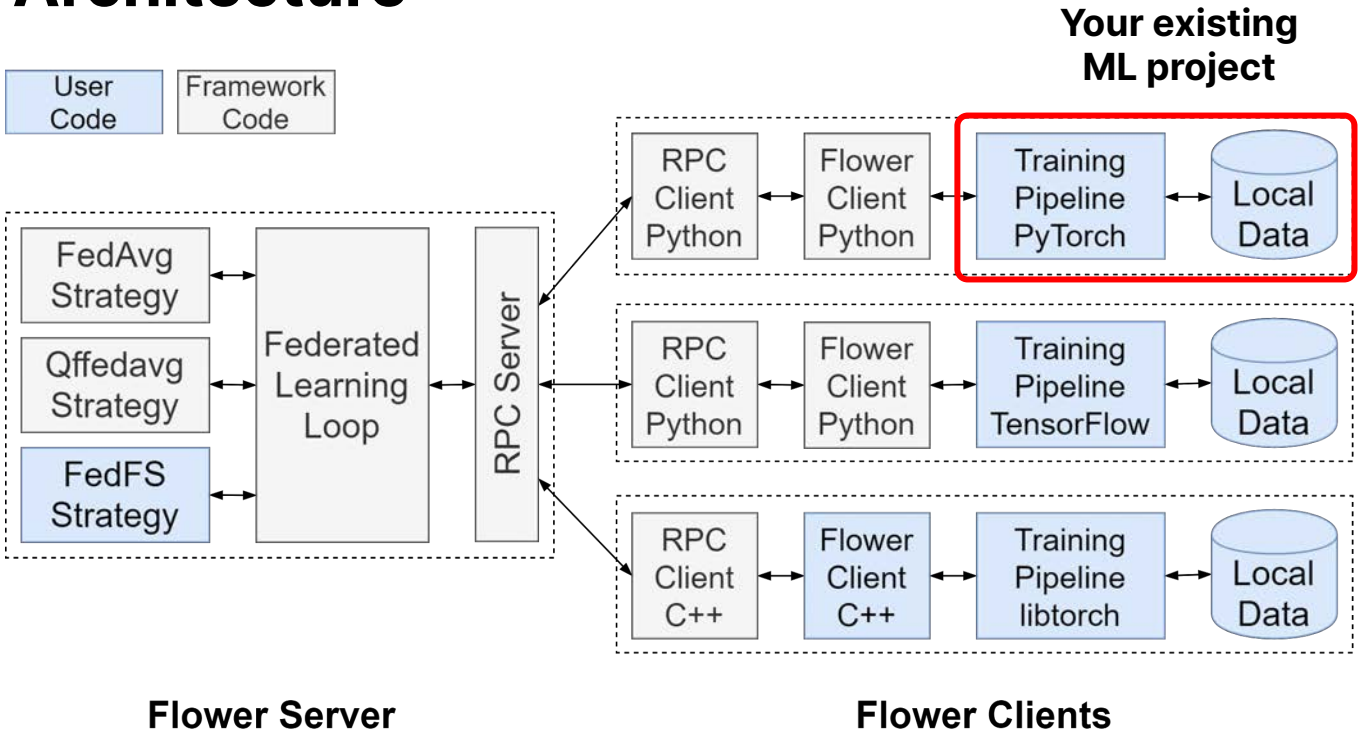




Flower
Modular Architecture



Flower
Modular Architecture

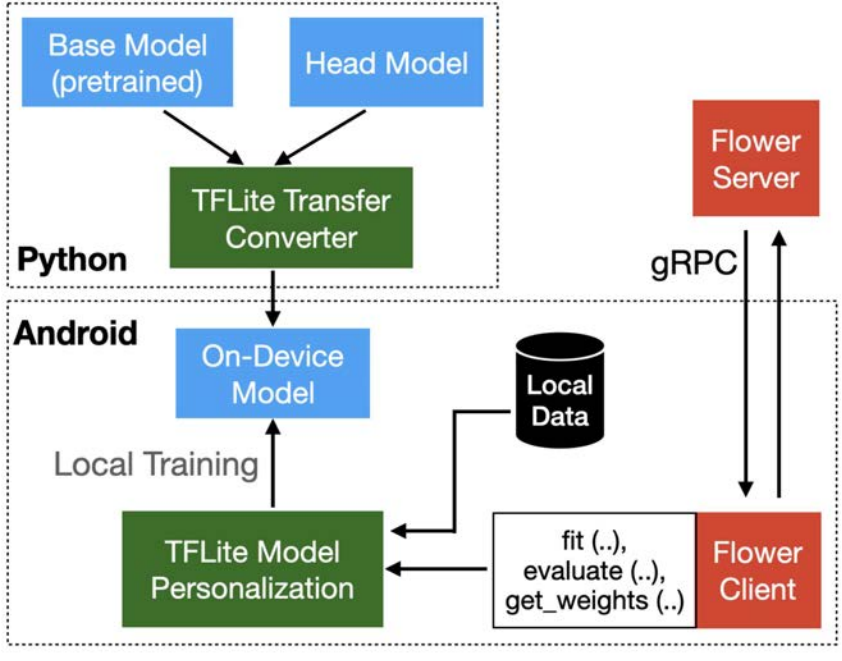


Flower Android

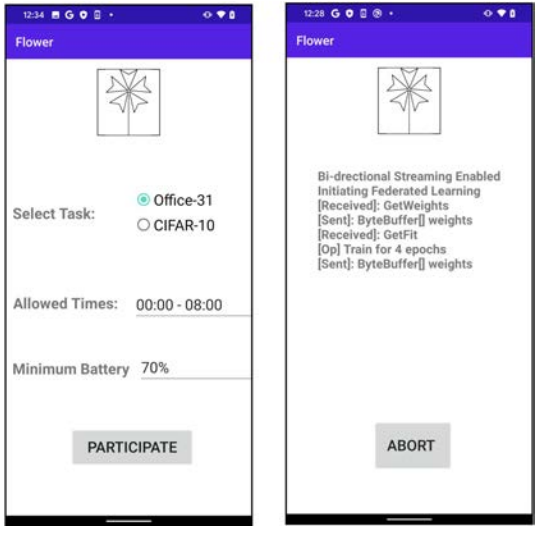
On-device training support on mobile devices is in infancy.

We leverage the Tensorflow Lite model personalization support on Android for Federated Learning.

Implementing three core functions to interface with Flower.



Flower Android



Flower Server deployed on EC2.

Flower Android Clients

- Personal Android smartphones
- Android phones and tablets in the AWS Device Farm.

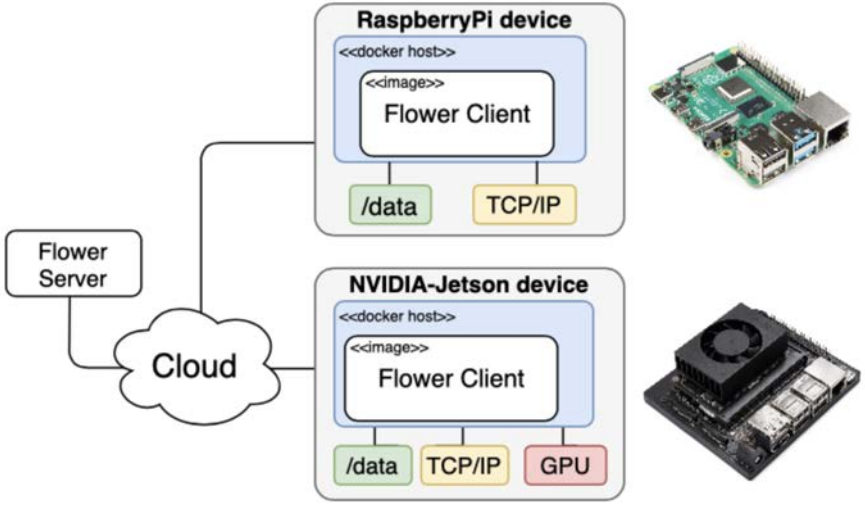
Device Name	Type	OS Version
Google Pixel 4	Phone	10
Google Pixel 3	Phone	10
Google Pixel 2	Phone	9
Samsung Galaxy Tab S6	Tablet	9
Samsung Galaxy Tab S4	Tablet	8.1.0

Flower Embedded

Flower clients implemented in Python for Raspberry Pi and Nvidia Jetson TX2.

Heterogeneous hardware, but same Flower client implementation.

Python and Android clients can co-exist



Evaluation

Runtime Costs of Federated Training

Local Epochs (E)	Accuracy	Convergence Time (mins)	Energy Consumption (kJ)
1	0.48	17.63	10.21
5	0.64	36.83	50.54
10	0.67	80.32	100.95

Performance on Nvidia Jetson TX2. 10 clients, 40 rounds

Dataset: CIFAR 10
Model: ResNet 18

No. of Clients (C)	Accuracy	Convergence Time (mins)	Energy Consumption (kJ)
4	0.84	30.7	10.4
7	0.85	31.3	19.72
10	0.87	31.8	28.0

Performance on Android smartphones. 5 epochs, 20 rounds.

Dataset: Office-31
Base Model: MobileNetV2
Head Model: 2 layer DNN

Computational Heterogeneity

FL Convergence time on CPU = 1.27x GPU

We can implement a device-aware FL strategy

Assign a cutoff time (τ) for each processor after which the device must send partial results.

Speed up convergence at the expense of some accuracy loss.

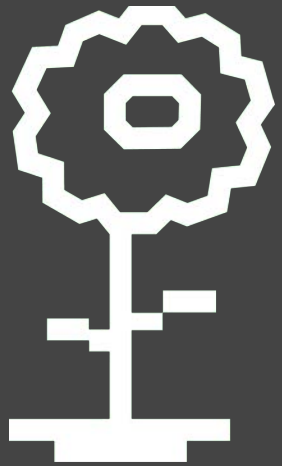
	GPU ($\tau = 0$)	CPU ($\tau = 0$)	CPU ($\tau = 2.23$)	CPU ($\tau = 1.99$)
Accuracy	0.67	0.67	0.66	0.63
Training time (mins)	80.32	102 (1.27x)	89.15 (1.11x)	80.34 (1.0x)

Performance on Nvidia Jetson TX2. 10 clients, 40 rounds

Dataset: CIFAR 10
Model: ResNet 18

Flower

https://flower.dev/



Akhil Mathur
Nokia Bell Labs and University of Cambridge
https://akhilmathurs.github.io/
@akhilmathurs