Towards a Modular FL Framework on Edge Devices

Roopkatha Banerjee[1,*], Prince Modi[1], Harsha Varun Marisetty[2,*], Manik Gupta[2] and Yogesh Simmhan[1]
[1]Indian Institute of Science, Bangalore, India [2]Birla Institute of Technology and Science Pilani, Hyderabad, India
{roopkathab, princemodi, simmhan}@iisc.ac.in, {p20200437, manik}@hyderabad.bits-pilani.ac.in

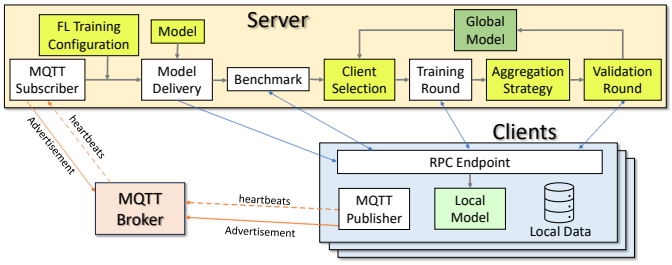

Figure 1: Architecture of FLOTILLA

I. Introduction

The proliferation of modern mobile and Internet of Things (IoT) devices has unleashed a torrent of sensor data generated on a continuous basis. However, transmitting this data from edge devices to the cloud for centralized model training incurs both time and network costs. *Federated Learning (FL)* [1] offers a privacy-preserving, decentralized training paradigm where models are trained on edge devices on their local data and the models aggregated centrally, without sharing the data.

Related Works and Gaps. FL frameworks have largely focused on the ML rather than the systems aspects. LEAF and TensorFlow Federated (TFF) simulate clients on a single machine, which limits the ability to study their system performance on real distributed devices. FLOWER [2] deploys clients on real edge devices, but has no provision for model delivery from the server to the clients.

Contributions. In this extended abstract, we introduce FLOTILLA, a modular, model-agnostic FL framework that supports synchronous client-selection and aggregation strategies, and FL model deployment and training on edge client clusters, while telemetry for advanced systems research.

II. Framework Design

Fig. 1 shows the architecture of FLOTILLA

1) Client Discovery and Benchmarking: The FL server uses a MQTT pub-sub mechanism for automatic discovery of clients. The clients publish their gRPC endpoints, a list of local datasets and model architectures, etc., and periodic heartbeats to a predetermined MQTT topic monitored by the server.

2) Model Training: The server uses the user-defined client selection strategy to initiate local training on the selected clients through a gRPC call, passing parameters on the epochs to train, the optimizer etc., and the global model weights. Once local training concludes, the clients return the new local model weights and accuracy metrics for the training round.

3) Performance Monitoring: System metrics like CPU, network utilization and disk IO are logged on both server and clients. After training, the server logs the training round metrics, like number of mini-batches etc., reported by the client, which can be used for system and model analysis.

III. Evaluation

1) Setup: We perform a preliminary evaluation of the initial FLOTILLA design on a cluster of 30 Raspberry Pi 4Bs, with the server hosted on a GPU workstation. We evaluate *AlexNet* on *EMNIST* dataset, which is divided into 30 partitions in an IID manner. The batch size is 16, the learning rate is 0.001, Adam is the optimizer, Cross Entropy is the loss function, and FedAvg [1] is used for model aggregation. We evaluate three client selection strategies in a round: *Default* (all available clients are selected), *Random Subset (RS)* ($\rho = 20\%$ of the available clients are selected randomly) and *Probabilistic High-Loss (PHL)* (clients are assigned probabilities in proportion to their validation losses and $\rho = 20\%$ chosen).

2) Results: In our analysis (plots omitted for brevity), *Default* is seen to converge the fastest, followed closely by *RS* and *PHL*. Since the data on each of the workers is IID, all locally trained models are aligned. Thus, in a setting with no stragglers, the *Default* strategy shows the fastest convergence, since it receives the most model updates in a round. Further, *PHL* is seen to perform similar to *RS*, since the validation losses of the clients will be very similar in an IID setting. Since the Pis are homogeneous and connected over Gigabit LAN, the client training times are similar and there are no stragglers. *PHL* shows a spike in training time in every other round owing to a short validation to find the validation loss on the clients. We report that the model takes 100 FL rounds to converge to an accuracy of 98.4%, with the average time taken per round being 375 seconds.

IV. Future Directions

As future work, we plan to support asynchronous and hierarchical FL strategies, and also allow easy deployment of decentralized FL training. We also plan to support more reliable training and recovery from failures of devices. More detailed experiments are planned on heterogeneous edges as well.

References

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, 2016.
[2] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. H. Li, T. Parcollet, P. P. B. de Gusmão *et al.*, "Flower: A friendly federated learning framework," 2022.