Demo: An Experimental Environment Based On Mini-PCs For Federated Learning Research

Felix Freitag*, Pedro Vilchez*, Lu Wei†, Chun-Hung Liu‡, Mennan Selimi§, Iordanis Koutsopoulos¶,
* Computer Architecture Department, UPC BarcelonaTech. Spain
† Department of Computer Science, Texas Tech University, Lubbock TX, USA
‡ Department of Electrical and Computer Engineering, Mississippi State University, Starkville MS, USA
§ Max van der Stoel Institute, South East European University. North Macedonia
¶ Department of Informatics, Athens University of Economics and Business, Athens, Greece

***Abstract*—There is a growing research interest in Federated Learning (FL), a promising approach for data privacy preservation and proximity of training to the network edge, where data is generated. Resource consumption for Machine Learning (ML) training and inference is important for edge nodes, but most of the proposed protocols and algorithms for FL are evaluated by simulations. In this demo paper, we present an environment based on distributed mini-PCs to enable experimental study of FL protocols and algorithms. We have installed low-capacity mini-PCs within a wireless city-level mesh network and deployed container-based FL components on these nodes. We show the deployed FL clients and server at different nodes in the city and demonstrate how an FL experiment can be set and run in a real environment.**

***Index Terms*—Federated Learning, Edge/cloud computing, Mini-PCs, Test-bed.**

I. Introduction

Federated Learning (FL) distributes the effort of training Machine Learning (ML) models over many distributed small nodes [1]. With FL, there is the opportunity to perform ML model training on edge devices, thus exploiting edge nodes' computing power and the emergence of lightweight ML frameworks such as TensorFlow Lite[1].

FL can unlock the obstacles of centralized ML approaches. An important feature is privacy preservation of local training data. Since only trained models rather than raw data are exchanged between server and clients, the characteristics of private local data are embedded in the trained models, and methods like differential privacy help to reduce what remains from the exposition of private data through these models [2]. Among popular FL-based applications are Apple's Siri, which leverages privacy-preserving FL[2].

While several variations of FL algorithms have been proposed and evaluated on different datasets, the practical aspects of FL are less well understood, while most of new ideas are validated only in simulation. Therefore, there exists a gap between the established theoretical knowledge and the answer to the question of what would be the building blocks of FL operating in real edge scenarios. However, there are a few works towards this direction, e.g. by Gao *et. al.* [3], in which FL experimentation is performed in controlled conditions with Raspberry Pi nodes.

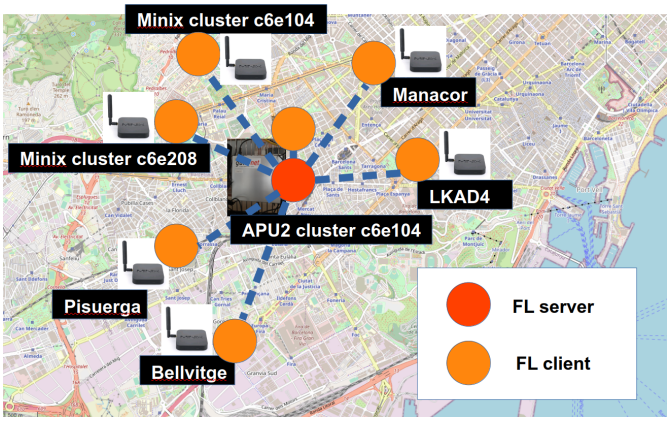

Figure 1. Commodity devices as testbed nodes deployed in the GuifiSants city mesh network with installed FL components.

In this demo paper, we present an experimental environment deployed within a wireless network, in which FL can be researched under real conditions. Figure 1 illustrates the testbed nodes when they are used for an experiment. Nodes are connected to the routers of a wireless mesh network called GuifiSants[3] in the city of Barcelona.

II. Experimentation Environment

The hardware used for testbed nodes consists of Minix mini-PCs[4] and PC Engines APU2[5]. The original operating system of these devices was replaced by Debian 10 Buster.

The GuifiSants wireless mesh network is part of the larger Guifi.net community network[6]. Thus, the testbed nodes (as part

[1]Deployed ML models on mobile and IoT devices. https://www.tensorflow.org/lite

[2]How Apple personalizes Siri without hoovering up your data: https://www.technologyreview.com/2019/12/11/131629/apple-ai-personalizes-siri-federated-learning/

[3]GuifiSants monitor. http://dsg.ac.upc.edu/qmpsu/index.php

[4]Minix NEO Z83-4, with Intel Atom x5-Z8350 processor and 4GB DDR3 RAM. https://minix.com.hk/products/neo-z83-4-pro

[5]PC Engines APU2 with AMD Embedded G series GX-412TC processor and 4 GB DDR3 RAM. https://pcengines.ch/apu2e4.htm

[6]https://guifi.net/

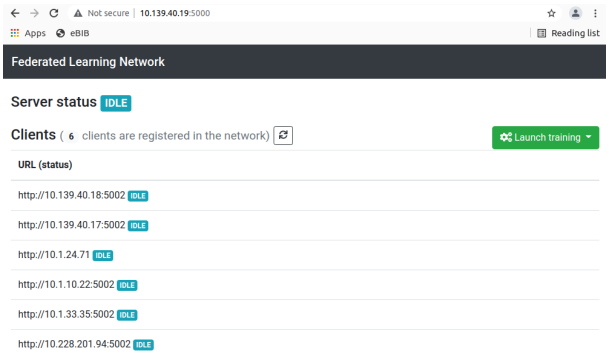

Figure 2. FL server web interface with the list of registered FL clients.

of Guifi.net) have routable IP addresses within Guifi.net assigned from the 10.0.0.0/8 network segment. Access to testbed nodes can be either remote from the public Internet for which we have created a Wireguard and VPN access, or by connecting locally to a Guifi node. We use an FL implementation with client and server components implemented in Python. The design of the implementation is modular, allowing one to experiment with different ML models or application cases. For the experimentation, we use an image classification task, for which a Convolutional Neural Network (CNN) is trained at each client. The code is packed in Docker images for the deployment at different nodes. We have installed a Docker registry and a Debian repository proxy within Guifi.net for nodes with limited or no Internet access. Thus, newly-built Docker images, which the experimenter creates for testing changes in protocols and algorithms, are pushed to the local Docker registry, and from there they can be pulled by any testbed node within Guifi.net.

The experimentation may study different parameters of the FL design space. One aspect can be the application level, e.g., analyzing the effects of different protocols and algorithms on inference accuracy. Another focus can be the architecture, in terms of client and server designs and their interactions. For edge scenarios, where nodes have limited bandwidth and computational capacities, the resource usage pattern of different FL designs and algorithms is important to understand. The experimentation environment may be controlled through the Web interface of the FL server (Figure 2).

Since the experimentation goals can be broad and diverse, we use more than one monitoring tool. For general long-term monitoring of experiments, we have implemented a Prometheus-Grafana solution. Fig. 3 shows a dashboard which monitors for an experiment the CPU, memory and bandwidth consumption of the FL server. Periodic patters can be observed, corresponding to the FL training rounds. For short-term experimentation, we have installed other open-source tools in the testbed nodes for measuring resource consumption and traffic of the FL component at the level of seconds.

III. Experimentation

Experimentation in this demo aims to show the capabilities and potential of the testbed environment in conducting FL

Figure 3. Grafana dashboard for FL node monitoring.

research experiments by showing:

1) Preparation of an experiment by choosing and registering a set of FL clients to the server and configuration options.
2) Running of an FL experiment on several distributed testbed nodes.
3) The steps for analysis of results and examples of detected behavior.
4) Our on-going work on extensions of the FL experimentation environment.

Acknowledgment

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871582 — NGIatlantic.eu and was partially supported by the Spanish Government under contracts PID2019-106774RB-C21, PCI2019-111851-2 (LeadingEdge CHIST-ERA), PCI2019-111850-2 (DiPET CHIST-ERA). The work of C.-H. Liu was supported in part by the U.S. National Science Foundation (NSF) under Award CNS-2006453 and in part by Mississippi State University under Grant ORED 253551-060702. The work of L. Wei is supported in part by the U.S. National Science Foundation (#2150486 and #2006612). I Koutsopoulos acknowledges support from the CHIST-ERA grant CHIST-ERA-18-SDCDN-004 (GSRI grant number T11EPA4-00056).

References

[1] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[2] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454–3469, 2020.

[3] Y. Gao, M. Kim, S. Abuadbba, Y. Kim, C. Thapa, K. Kim, S. A. Camtep, H. Kim, and S. Nepal, "End-to-end evaluation of federated learning and split learning for internet of things," in *2020 International Symposium on Reliable Distributed Systems (SRDS)*, 2020, pp. 91–100.