FLEX: Trading Edge Computing Resources for Federated Learning via Blockchain

虽然提到可以交易数据和计算资源，但是文章只涉及了交易FL模型，即server根据定价寻找合适的client model进行aggregate因为只传输model ID，所以也不涉及模型存储问题。区块链采用的ETH私链

Yang Deng[1], Tao Han[1], Ning Zhang[2]
[1]University of North Carolina at Charlotte, USA,
[2]University of Windsor, Canada
{*ydeng3, tao.han*}*@uncc.edu, ning.zhang@uwindsor.ca*

***Abstract*—Federated learning (FL) algorithms provide privileges in personal data protection and information islands elimination for distributed machine learning. As an increasing number of edge devices connected in networks, we still see a lot of computing resources and data remaining underutilized and there is no platform for users to trade FL tasks. In this demonstration, we propose a blockchain-based federated learning application trading platform called FLEX, on which users can buy and sell computing resources for training machine learning models with no sacrifice of data privacy. We design FLEX in a highly distributed and scalable manner. We separate the data plane and control plane in the platform. In FLEX, trading mechanisms and FL algorithms are deployed in smart contracts of the blockchain. Control messages and trading information are well protected in the blockchain. With FLEX, we realize a distributed trading platform for executing FL tasks.**

***Index Terms*—federated learning, blockchain**

I. Introduction

Deep neural networks (DNN) have exhibited the capacities in a variety of applications. For training a general DNN model, one of the most important part is collecting data for training. However, with increasing attention to personal data protection, General Data Protection Regulation (GPDR) and other rules put restrictions of acquiring personal data. Federated Learning (FL) provides a methodology allowing users contributing to model training without sharing local data with others. In a FL application for training DNN, the datasets are distributed among local nodes for training local models which are aggregated in the central node to obtain the global model [1]. During the FL model training, training dataset is maintained in each node locally while only model updates are shared between local nodes and the central node.

In the context of edge computing, with the emerging of edge devices such as smart phones, IoT devices, vehicles, these edge devices are capable of processing certain DNN training tasks and holding local data which could be utilized in FL model training. The fact is that users become aware of how valuable their data are. But the value of their local resources are far beyond fully utilization in FL. One of the reasons is the lack of an open exchange platform for FL applications, where users can sell the remaining computing resources and personal data for local model training. In return, they get paid from the buyers who need models trained. In this way, users can trade the computing resources and local datasets in their edge devices with rewards given by model requester.

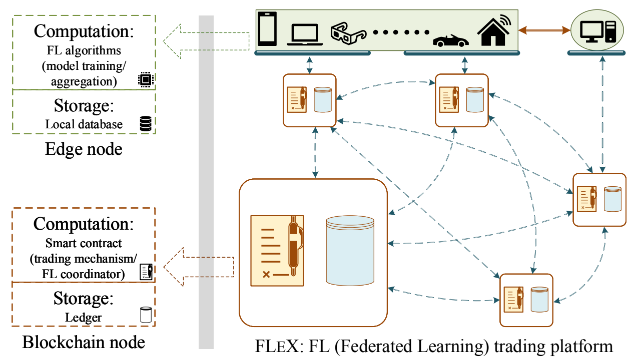

Fig. 1. FLEX framework.

FL applications involve local training across edge devices and global aggregation. To design a trading platform for FL applications, the following challenges need to be addressed. First, the platform needs to verify the users' authenticity before joining the FL tasks. Second, the platform can trace the transactions among edge devices. Third, the trading platform needs to guarantee the consistency of the model updates delivered. Fourth, to support distributed application like FL, the platform is expected to have the ability to support the scalability for distributed applications.

We address the requirement and challenges of trading FL tasks in this demonstration. Based on these challenges and the distributed features in FL applications, we use blockchain technique [2] in our platform design. We propose the first blockchain-based FL trading platform, FLEX. We demonstrate the prototype of FLEX and illustrate the complete trading workflow from resources querying to FL model delivering.

II. Framework of FLEX

As a trading platform for FL applications, the design goal of FLEX is to allow edge devices to query FL models from other devices. In the meantime, devices can access the platform to trade their computing resources and training data for rewards.

The framework of FLEX is shown in Fig. 1. The whole trading platform consists of blockchain nodes, devices nodes for local training, and server nodes for global aggregation. For the nodes in blockchain, they form a peer-to-peer (P2P) network to provide the computing resources for all functions in blockchain, such as query and change the states in smart contract. As shown in Fig. 1, the local training nodes and global aggregation nodes are connected via the blockchain net-

IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) | 978-1-6654-0443-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/INFOCOMWKSHPS51825.2021.9484628

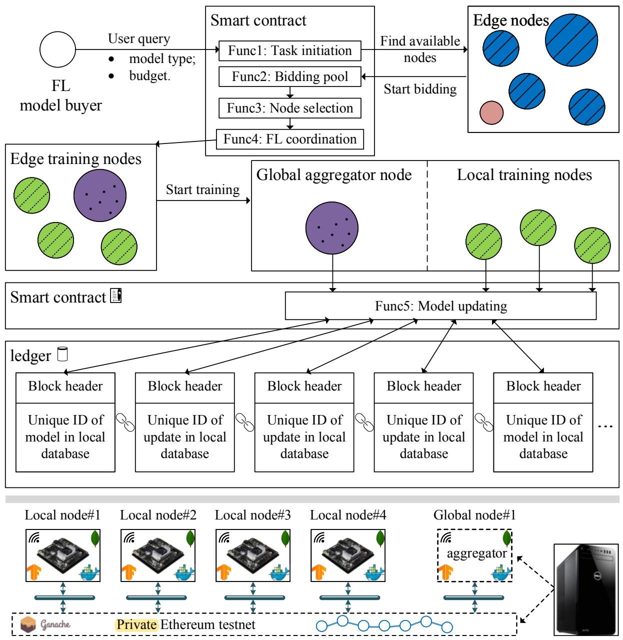

Fig. 2. User query workflow and prototype.

work. The local training nodes can be heterogeneous devices, such as mobile phones, IoT devices, and connected vehicles. When these edge devices are available, they may be selected for training FL models with their own local datasets. Different from centralized exchange center, we deploy control functions and trading policies in smart contracts, which are executed across the nodes in the blockchain.

In our prototype as shown in Fig. 2, we use Nvidia Jetson TX2 as local training node. We run a global node along with a Ganache private Ethereum testnet in the workstation equipped with i7-8700 CPU and RTX 2080 GPU. Edge devices access to the blockchain network via a Linksys WRT1900AC router. The global aggregation and local training are implemented with Tensorflow and executed in Nvidia dockers. These nodes interact with the blockchain nodes via Web3 API in Python.

As shown in the Fig. 1, we have two separate paths in the platform: control plane and data plane. Control plane includes trading functions and FL coordinating functions in smart contracts in the blockchain. In the data plane, each node has a local database for storing either local model updates or global model updates. We provide details in how a user's query will be processed in the following section.

III. User Query Processing Workflow in FLeX

In this section, we provide more details about how FLeX process a user's query of FL model training. Fig .2 shows the detailed workflow of trading and FL training in FLeX.

A. *Trading mechanism in* FLeX

We design FLeX as a platform for users to get profits from contributing to FL model training and for the other users to buy computing resources and users' personal data (without sharing) to obtain FL models. In FLeX, the user who intends to buy a well-trained model through the platform first sets a total budget limit for a certain FL training task. The nodes in the blockchain will verify whether the budget is eligible for a FL task. Once being verified, the functions deployed in smart contract in each node in the blockchain network will notify current available edge nodes to start bidding to join the FL task. The values of bids from edge nodes rely on the computing resources they can provide and the rewards they hope to get. After the function in smart contract receives all bids from edge nodes, the node selection function is triggered to select a set of nodes for the FL task. Next, the edge nodes selected will get FL training instructions from the FL coordination function in smart contract. At last, edge nodes obtain rewards based on the actual contributions to the FL model and the user get the well-trained model with certain payment within the budget.

B. *FL model training in* FLeX

After edge node selection, the model training workflow starts. Edge nodes are divided into two categories: one global server node and several local training nodes. At the beginning, the global node specifies the structure of the model and instructs all local nodes to maintain the consistency of the model structure. In each round of training, each local node updates the global model with its own dataset. Once it is completed, the local model update is saved into local database. Inspired by using third reliable storage to save models updates in [3], we choose to upload the unique ID of the update in local databases to the state in the smart contract. After the addresses are verified, a new block will be generated. The global nodes can access the local databases in the data plane to aggregate a new global model from all local model updates. Then the latest global model will be saved into the database located in the global server node. Similarly, the unique ID of the latest model address in the database will be uploaded in the smart contract and verified in the blockchain, which is accessible by local nodes for the next round of model training. To eliminate the latency introduced by transmitting large model weights in blockchain, we only transmit the unique ID of the model weights in blockchain network.

需要一个global server做aggregate

更新后的模型不会传输给client来减少通讯开销

只传输模型的ID

IV. Conclusion and Future Work

In this demonstration, we first analyze the necessity of building a trading platform for federated learning applications. Then, we discuss the main challenges in the design. Finally, we address these challenges by prototyping the new blockchain-based FL trading platform, FLeX. For future work, we will design the algorithms to optimize the bid strategy for more profits for participants and to balance the speed of training and the usage of computing resources in the platform.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, vol. 54, 2017, pp. 1273–1282.

[2] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, no. 37, 2014.

[3] Y. Li, C. Chen, N. Liu, H. Huang, Z. Zheng, and Q. Yan, "A Blockchain-based Decentralized Federated Learning Framework with Committee Consensus," *arXiv e-prints*, Apr. 2020.