Jupiter: A Modern Federated Learning Platform for Regional Medical Care

Ju Xing
Tsinghua university
xingj15@mails.tsinghua.edu.cn

Zexun Jiang
Tsinghua University
_jiangzx14@mails.tsinghua.edu.cn

Hao Yin
Tsinghua University
h-yin@mail.tsinghua.edu.cn

***Abstract* — In this paper we propose Jupiter, an easy-to-use, secure and high-performance federated learning platform for regional medical care. Jupiter provides innovative programming abstractions to make data tunning more efficient as well as high-performance infrastructures to accelerate secure aggregations of parameters. Jupiter employs a stateful design with a bunch of optimizations and leverages popular techniques like SDN, DPDK and Intel SGX. The experiments show that with a low memory footprint, the throughput of single aggregator can reach 300MB/sec(with slice size fixed to 64KB), and the aggregation primitive we built can process 11k aggregations per second.**

Keywords — federated learning; programming abstraction; performance; Intel SGX

Federated Learning[4] is an emerging technology to release the intrinsic value of data while giving full respect to data sovereignty. It has been initially practiced in various domains like finance, audit etc. However, building a federated learning platform in the scenario of regional medical care faces some challenges. First, data tunning in the training process is in high demand. A continued workflow should be fulfilled to promote efficiency and cost-reduction. Traditional deep learning frameworks often focus on the convenience of model constructions with the assumption that data and model are collocated, thus lacks the care of continued development; Second, parameters are usually aggregated by a third-party. Neither intermediate results or final results should be revealed to this party. Current works[2] which are based on secret sharing still can't prevent the third-party from seeing the final results; Third, federated learning is both data-intensive and compute-intensive. With the design of high-performance in the first place will make the system more practical for business workload. Here we propose Jupiter as our solution to the federated learning platform in regional medical care and solves these challenges accordingly.

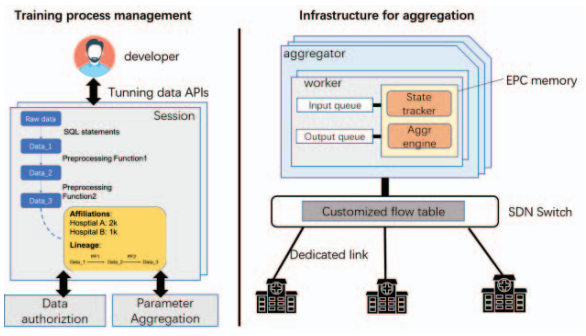

Fig. 1 Jupiter platform

Fig. 1 shows core features of Jupiter Platform. In order to support data tunning, Jupiter provides Federated Learning data type(FL_data) and session concept. The assembly of federated data and further transformation are tracked internally with this data type. Besides, FL_Data exposes rich APIs for operations like augmentations, eliminations and filters etc. Therefore, developers can efficiently amend their choices without restart a design process and can easily share logical logical datasets. All FL_Datas in a continued development process are managed with a session, which is identified by a token when the developer first submit her design. For the purpose of secure aggregation, we construct aggregators based on Intel SGX[3], an popular TEE technology, to guarantee the confidentiality of parameters end-to-end. Unlike traditional federated learning platform[1] serving mobile applications, Jupiter however takes a stateful design on top of dedicated links between aggregators and hospitals. As a results, parameters are aggregated in a streaming style, and all the related states are tracked and preserved inside the encrypted memory(EPC). Those states can be utilized for failure recovery. For high-performance consideration, our design is three-fold. First, Jupiter slices parameters into fixed-size chunks(e.g. 64KB) to overlap communications and computations in the learning process. Besides, it is noted that chunks are more suitable in dedicated link case because the bandwidth is usually highly limited. Second, to deal with fixed-size parameter chunk aggregation, Jupiter implements an aggregation engine leveraging SIMD characteristics of CPU, which can dramatically boost the performance. Third, in Jupiter, a two-level routing mechanism is used for delivery of parameters. Since processing chunk routing and duplications in the application level will incur huge overhead, Jupiter instead uses customized ethernet packet and encoding application meta data directly into packet field for routing. The flow table inside SDN switch can be programmed with application information instead.

REFERENCES

[1] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecny, S. Mazzocchi, H. B. McMahan et al., "Towards federated learning at scale: System design," arXiv preprint arXiv:1902.01046, 2019.

[2] A. Segal, A. Marcedone, B. Kreuter, D. Ramage, H. B. McMahan, K. Seth, K. Bonawitz, S. Patel, and V. Ivanov, "Practical secure aggregation for privacy-preserving machine learning," 2017.

[3] V. Costan and S. Devadas, "Intel sgx explained." IACR Cryptology ePrint Archive, vol. 2016, no. 086, pp. 1–118, 2016.

[4] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 10, no. 2, p. 12, 2019.

978-1-7281-6981-1/20/$31.00 ©2020 IEEE
DOI 10.1109/JCC49151.2020.00012