

ParSecureML: An Efficient Parallel Secure Machine Learning Framework on GPUs

Zheng Chen[◊], Feng Zhang[◊], Amelie Chi Zhou[★],
Jidong Zhai⁺, Chenyang Zhang[◊], Xiaoyong Du[◊]

[◊]Renmin University of China

[★]ShenZhen University

⁺Tsinghua University

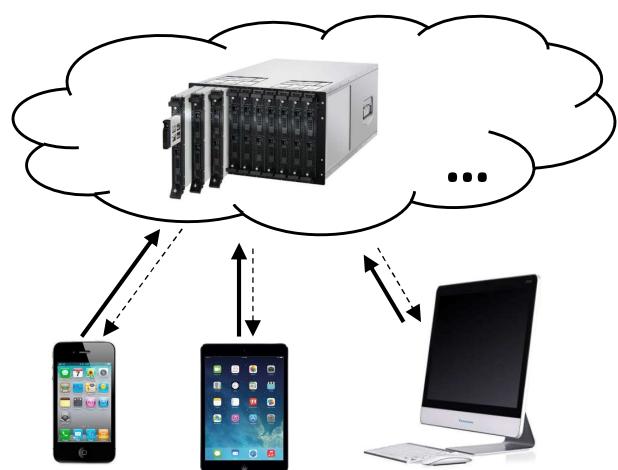


Outline

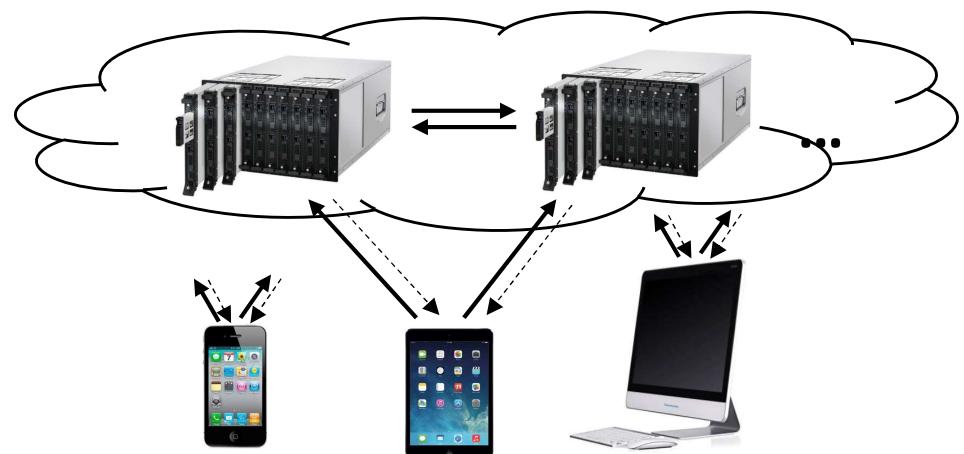
1. Background
2. Motivation
3. Basic Idea
4. Challenges
5. ParSecureML
6. Evaluation
7. Source Code at Github
8. Conclusion

1. Background

- Secure Machine Learning



(a) Typical machine learning process.



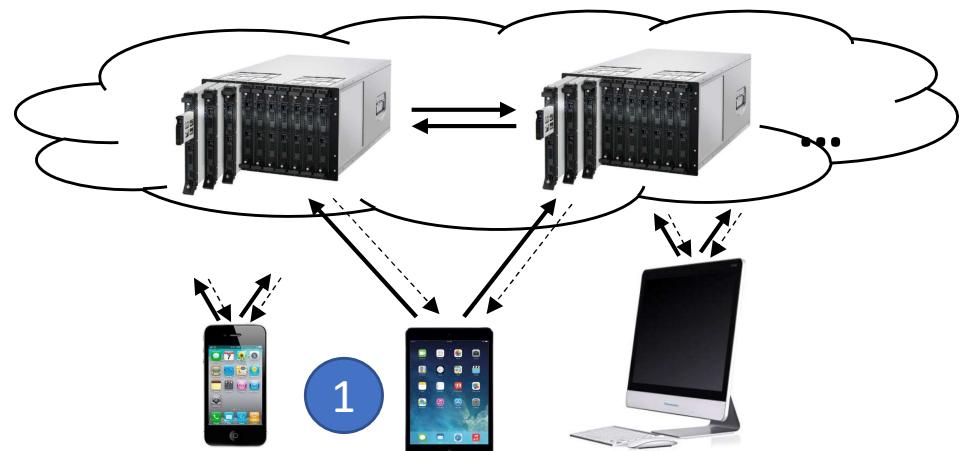
(b) Machine learning process with two-party computation.

1. Background

- Secure Machine Learning



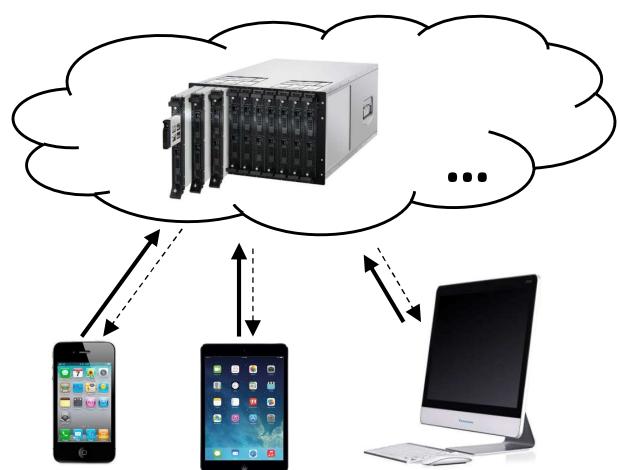
(a) Typical machine learning process.



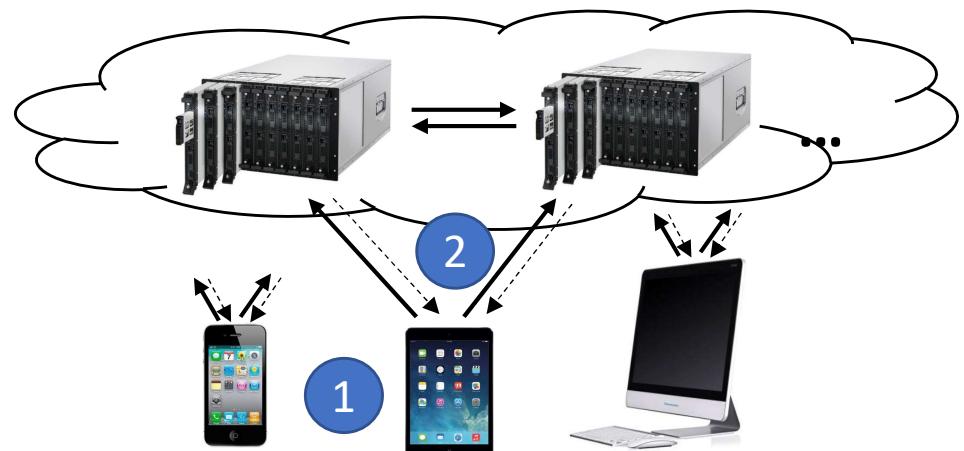
(b) Machine learning process with two-party computation.

1. Background

- Secure Machine Learning



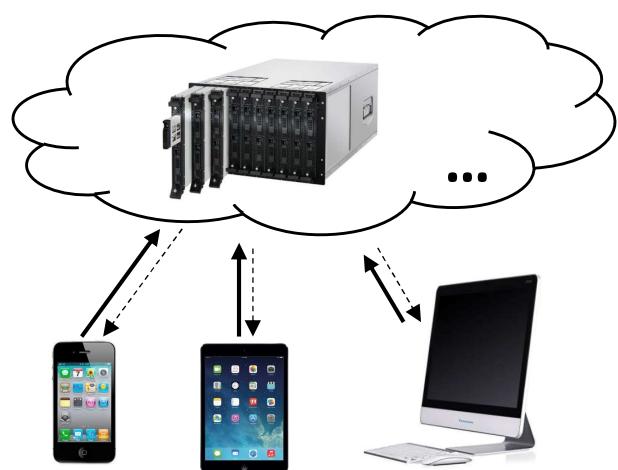
(a) Typical machine learning process.



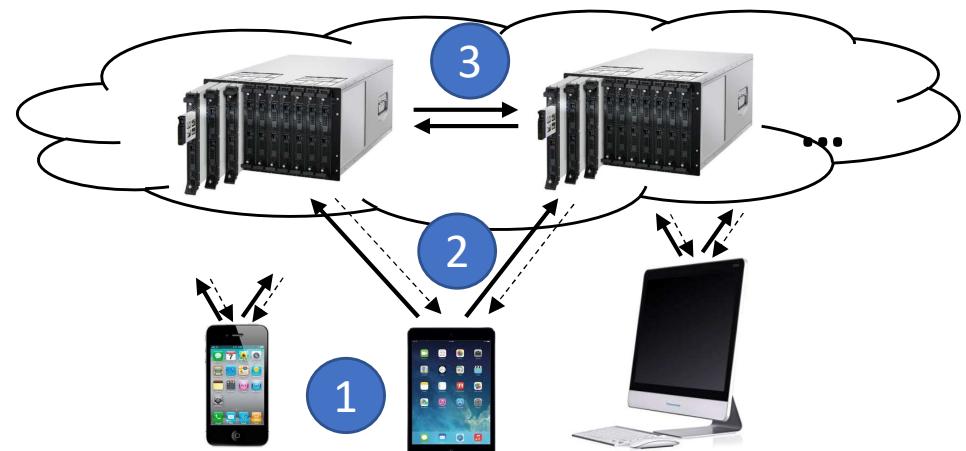
(b) Machine learning process with two-party computation.

1. Background

- Secure Machine Learning



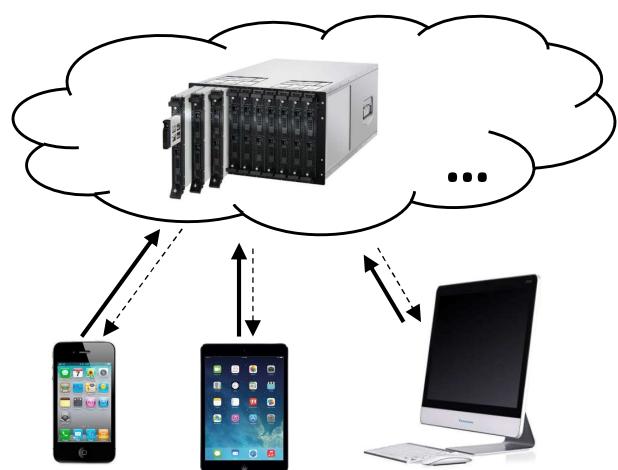
(a) Typical machine learning process.



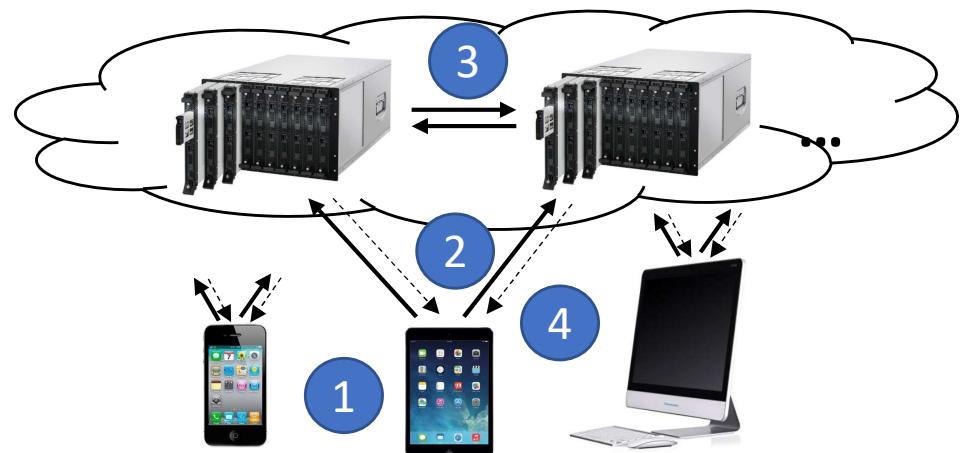
(b) Machine learning process with two-party computation.

1. Background

- Secure Machine Learning



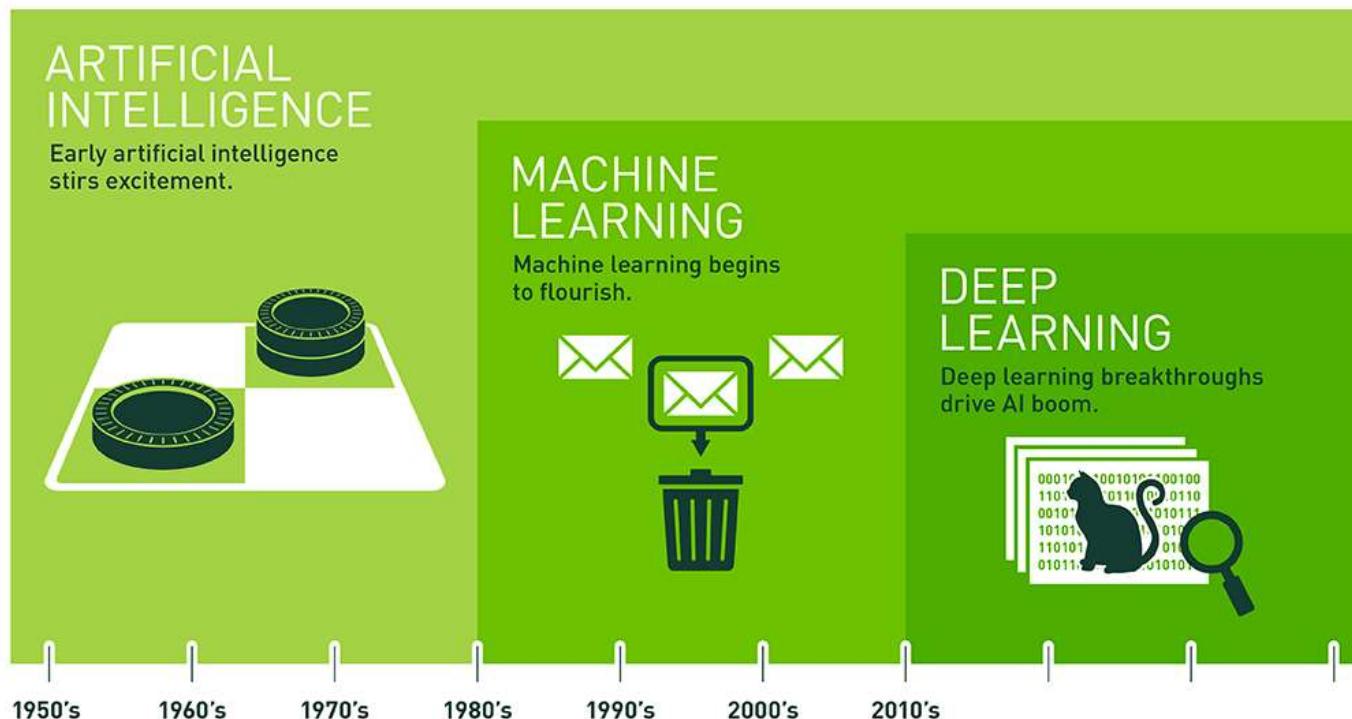
(a) Typical machine learning process.



(b) Machine learning process with two-party computation.

1. Background

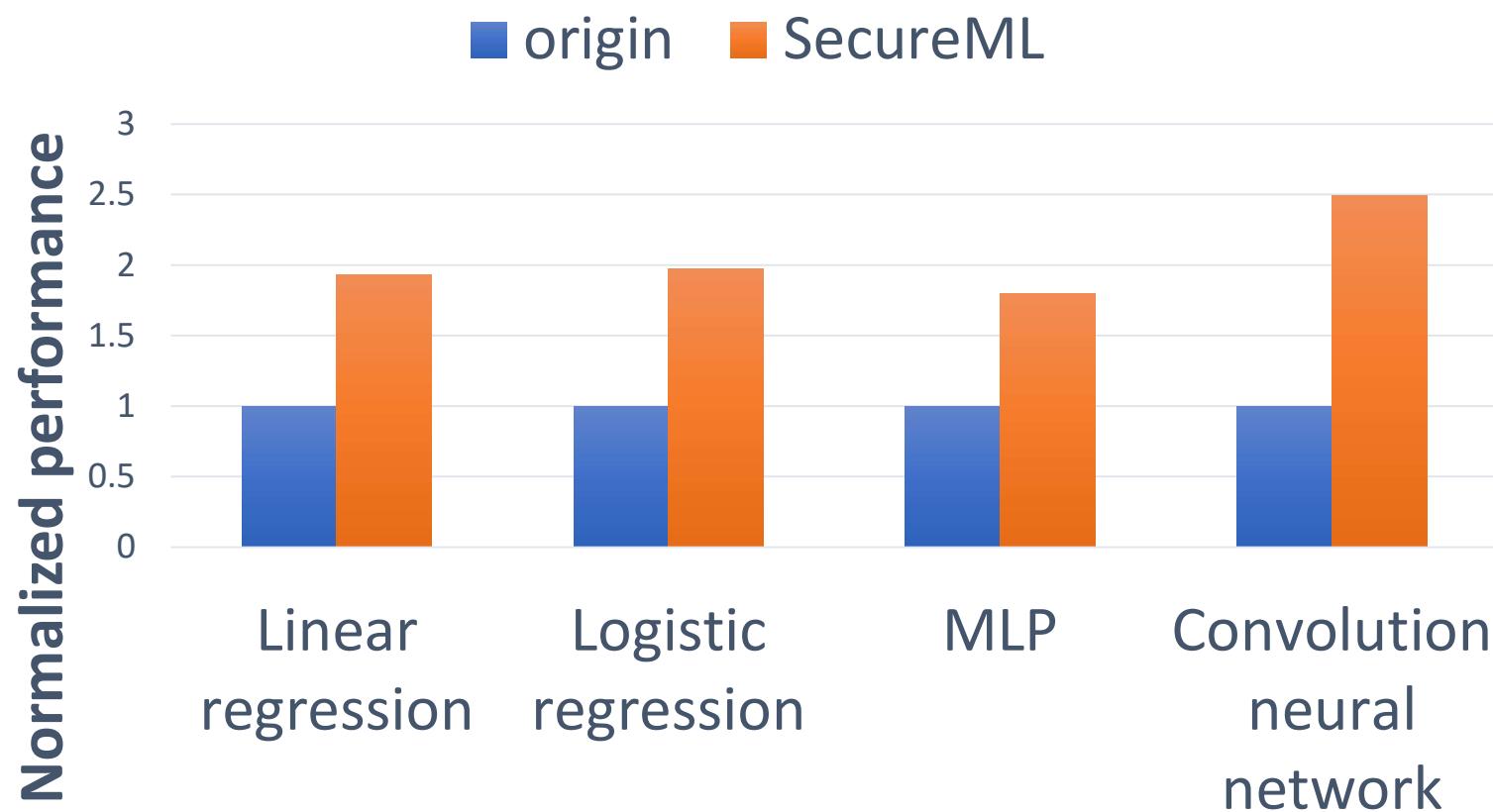
- GPU Acceleration



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

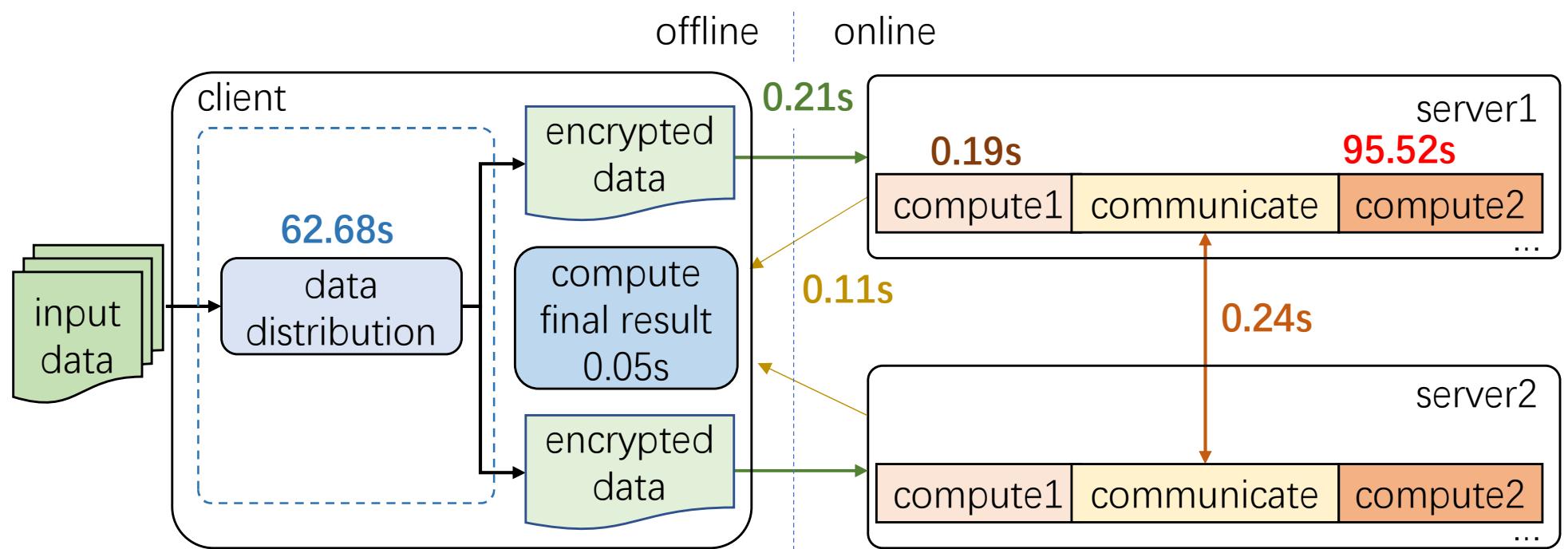
2. Motivation

- Performance Degradation



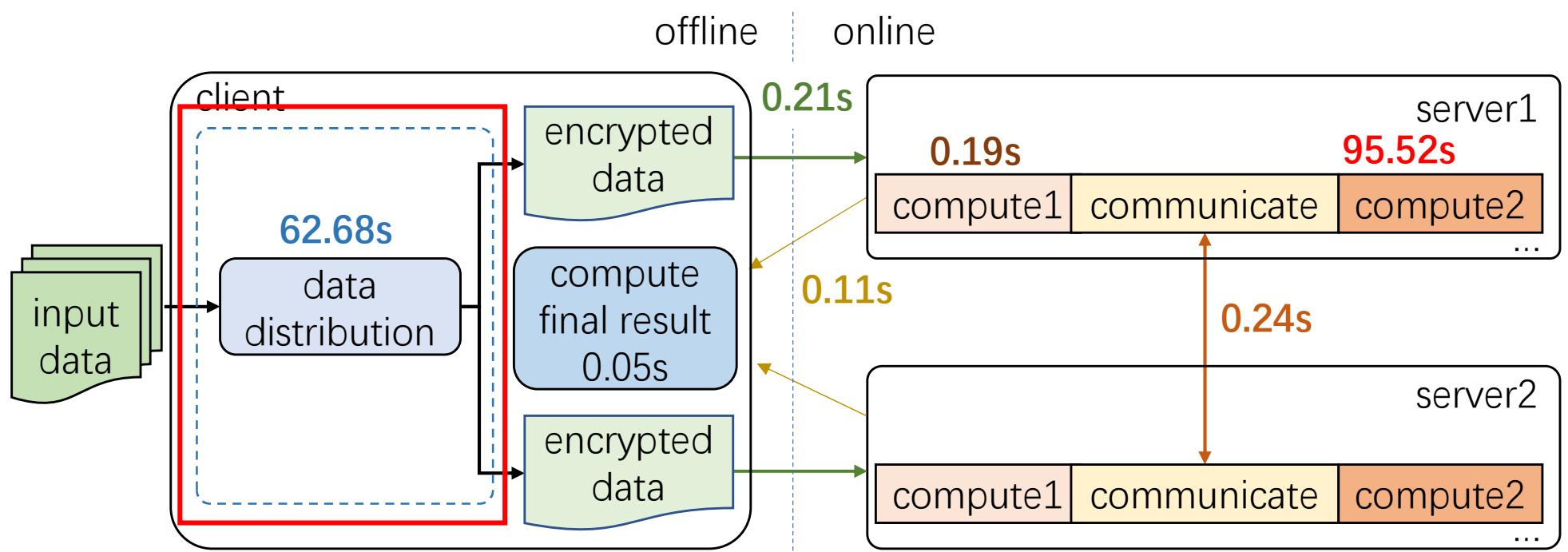
2. Motivation

- Time Breakdown for two-party computation



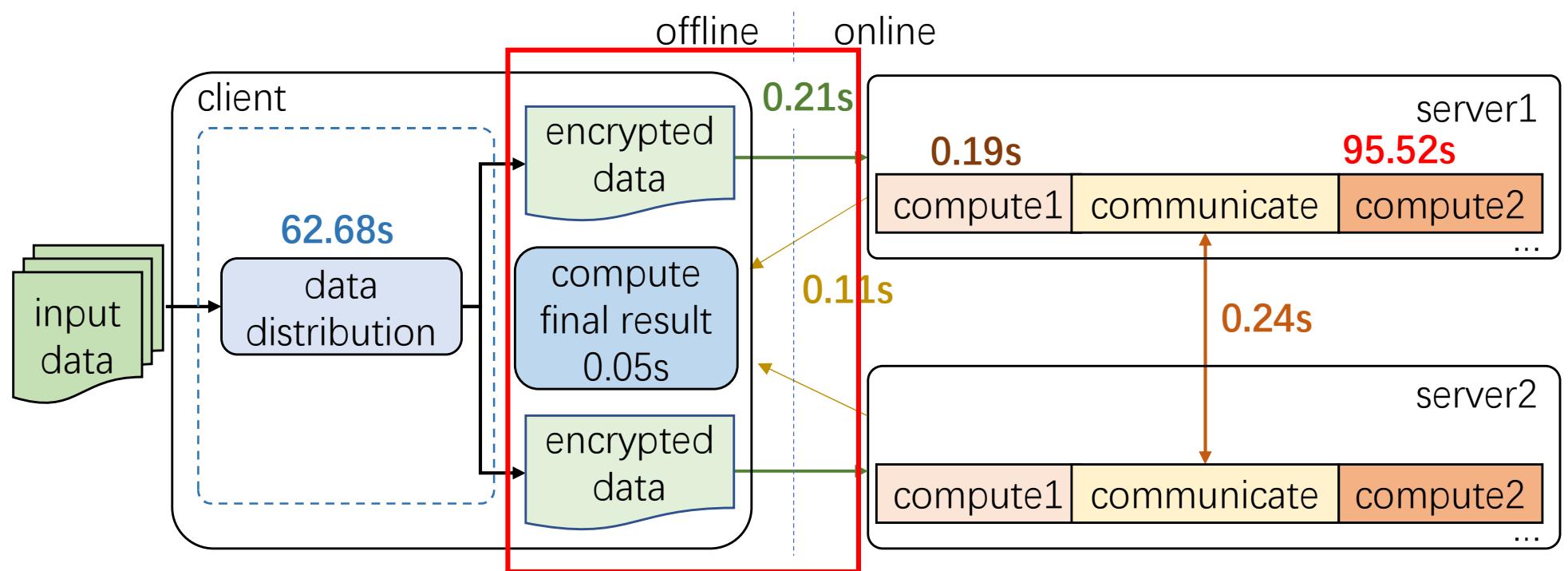
2. Motivation

- Time Breakdown for two-party computation



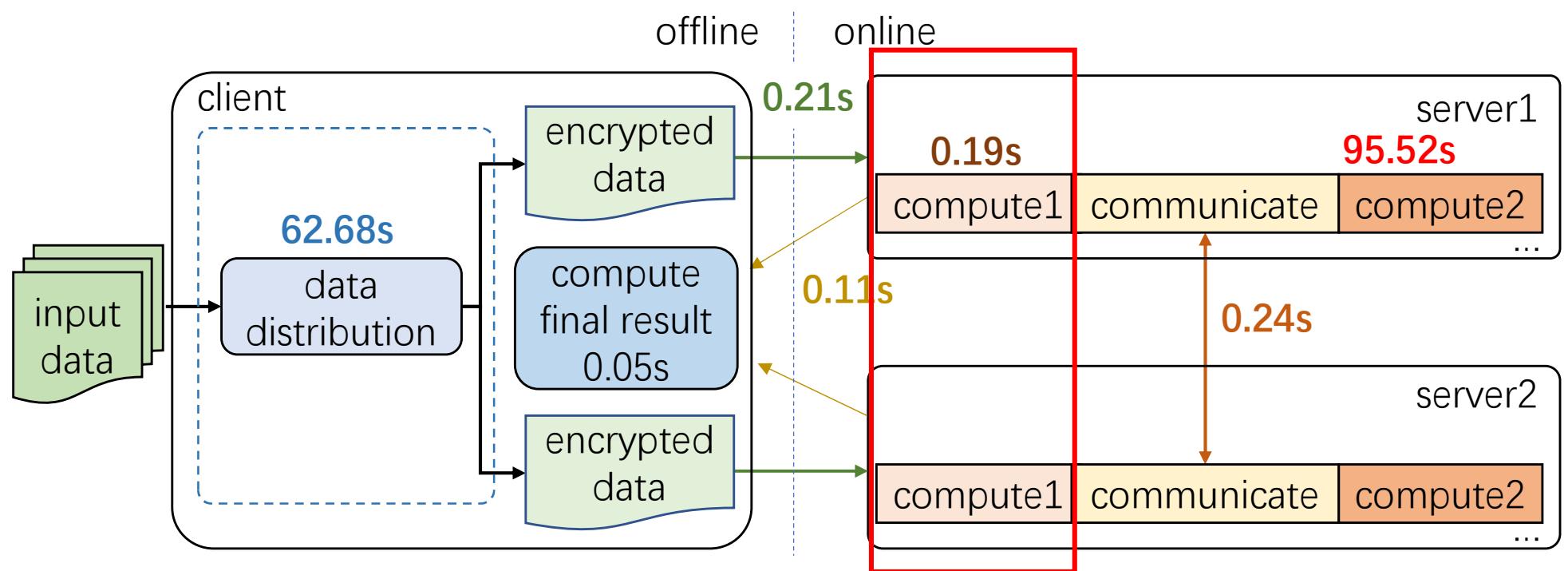
2. Motivation

- Time Breakdown for two-party computation



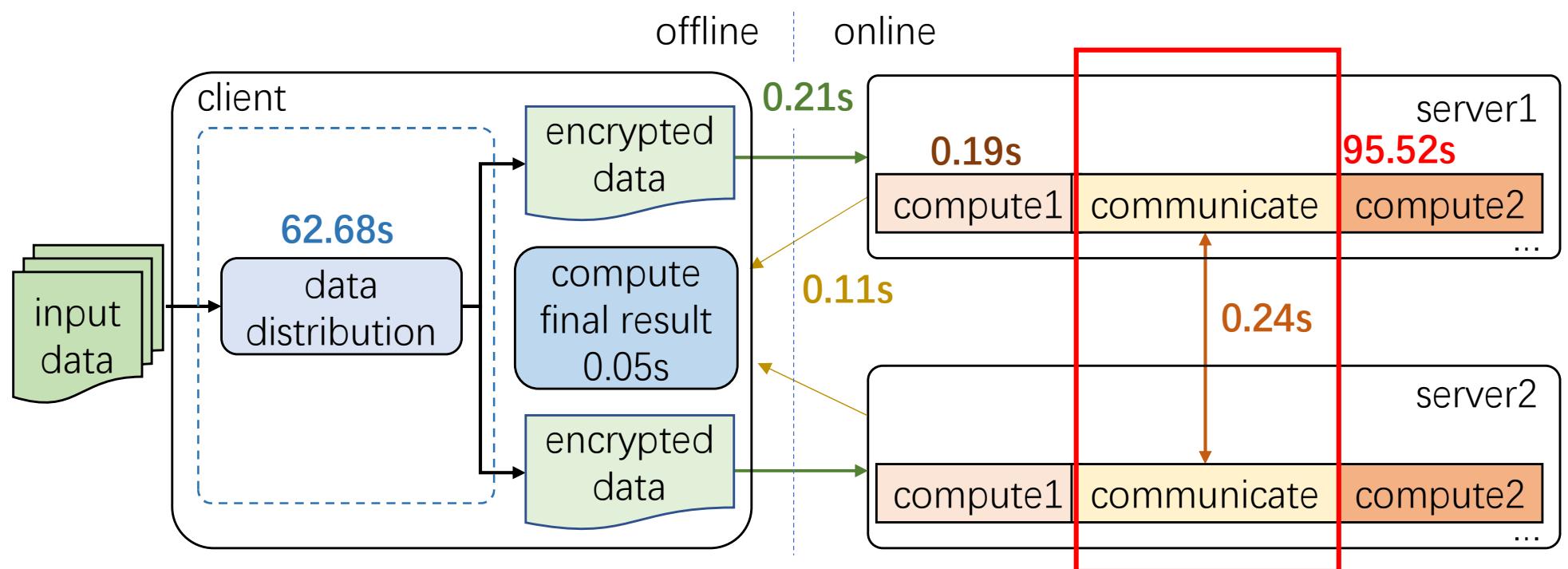
2. Motivation

- Time Breakdown for two-party computation



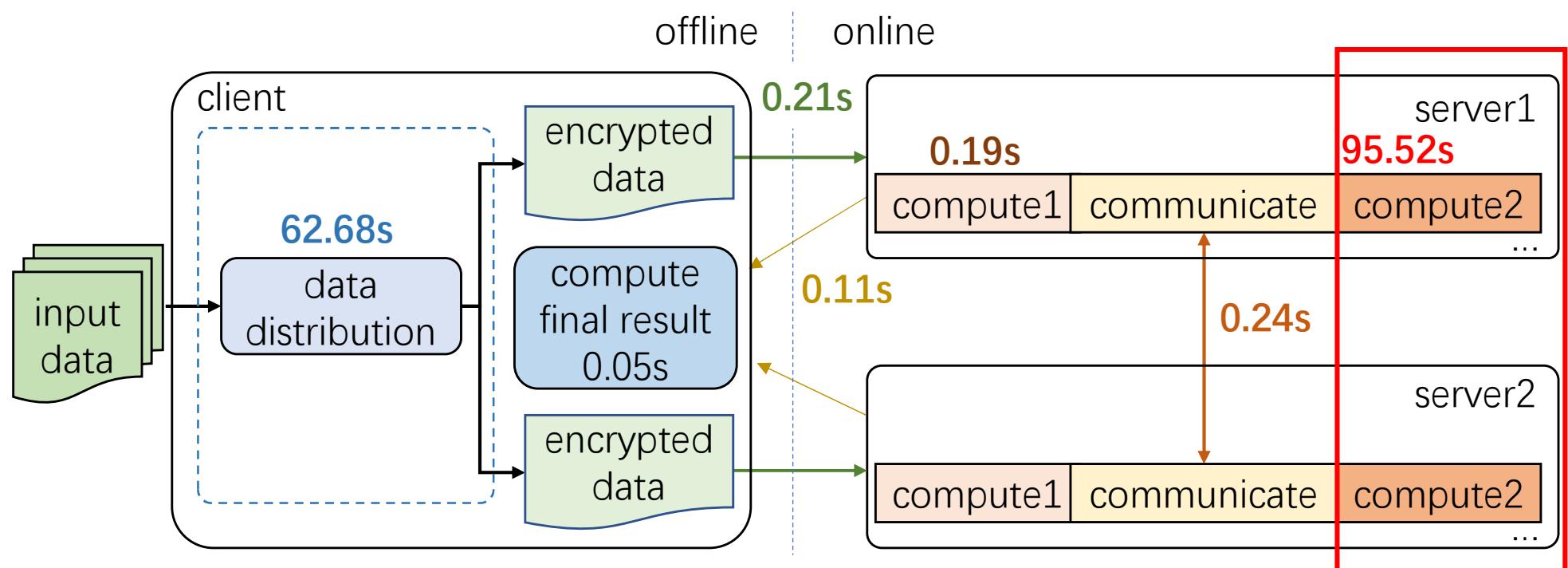
2. Motivation

- Time Breakdown for two-party computation



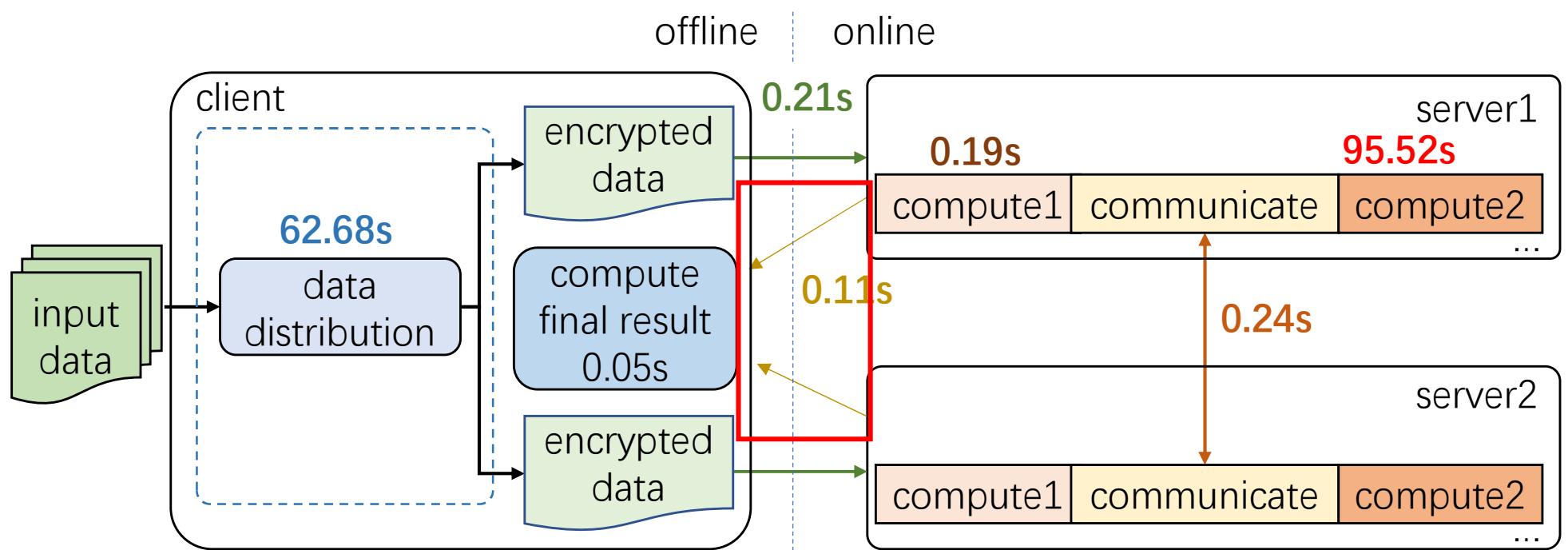
2. Motivation

- Time Breakdown for two-party computation



2. Motivation

- Time Breakdown for two-party computation



3. Basic Idea

- A GPU-based two-party computation that considers both the GPU characteristics and features of two-party computation shall have better performance acceleration effects.
- Challenges
 - Challenge 1: Complex triplet multiplication based computation patterns
 - Challenge 2: Frequent intra-node data transmission between CPU and GPU
 - Challenge 3: Complicated inter-node data dependence

4. Challenges

- Challenge 1: Complex triplet multiplication based computation patterns

$$C = A \times B \quad (1)$$

$$Z = U \times V \quad (2)$$

$$U = U_0 + U_1, V = V_0 + V_1, Z = Z_0 + Z_1 \quad (3)$$

$$E_i = A_i - U_i, F_i = B_i - V_i \quad (4)$$

$$E = E_0 + E_1, F = F_0 + F_1 \quad (5)$$

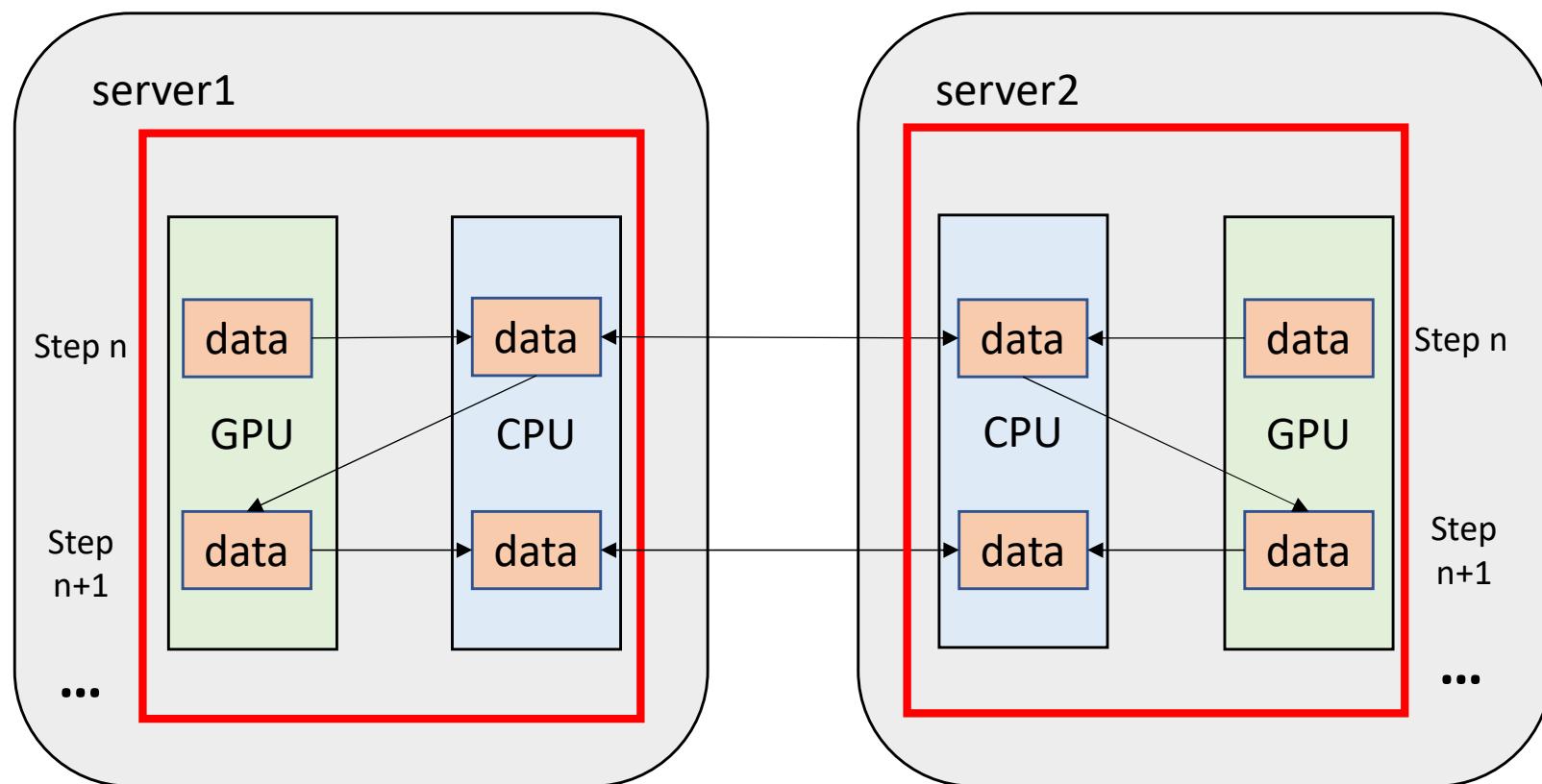
$$C_i = (-i) \times E \times F + A_i \times F + E \times B_i + Z_i \quad (6)$$

...

...

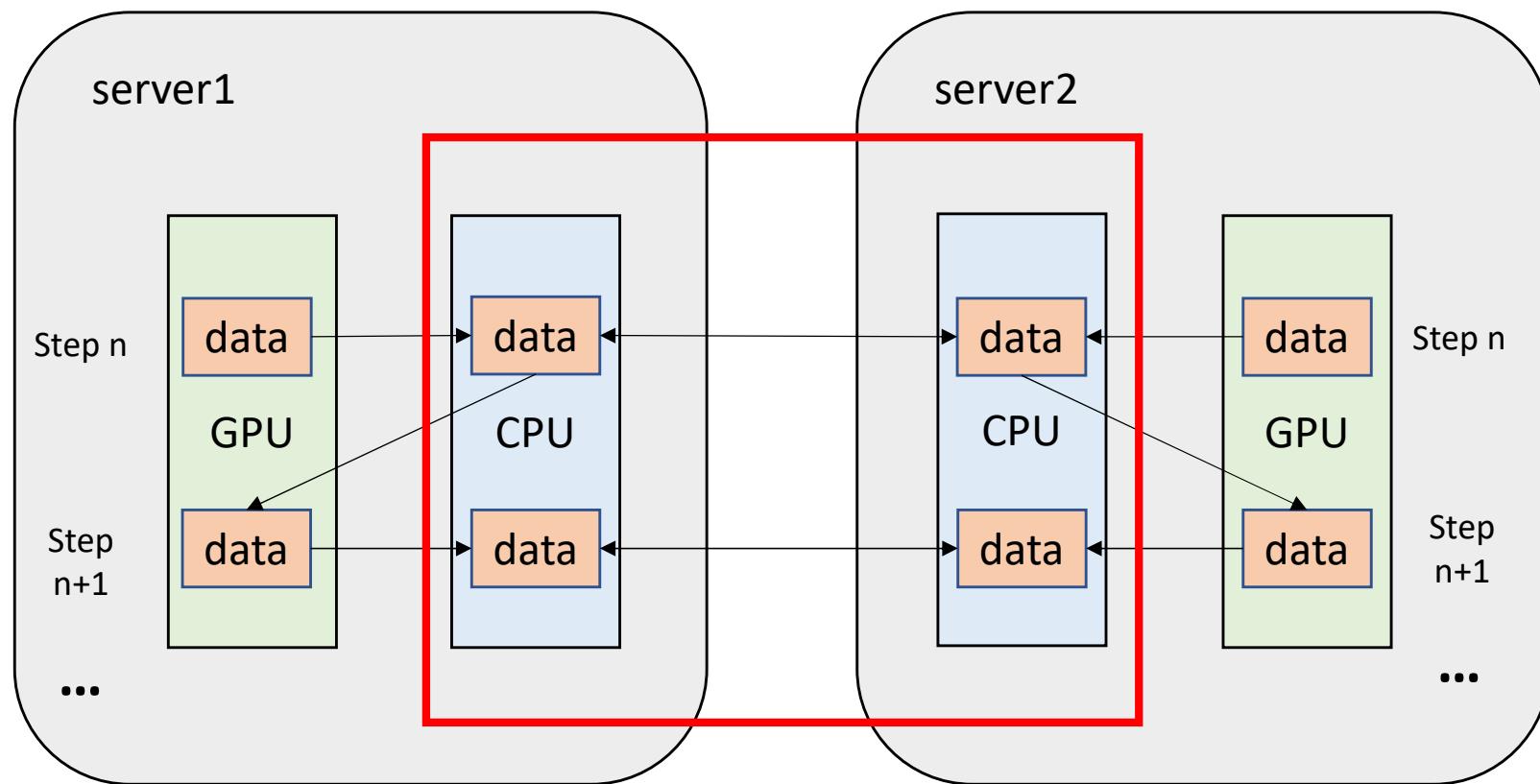
4. Challenges

- Challenge 2: Frequent intra-node data transmission between CPU and GPU



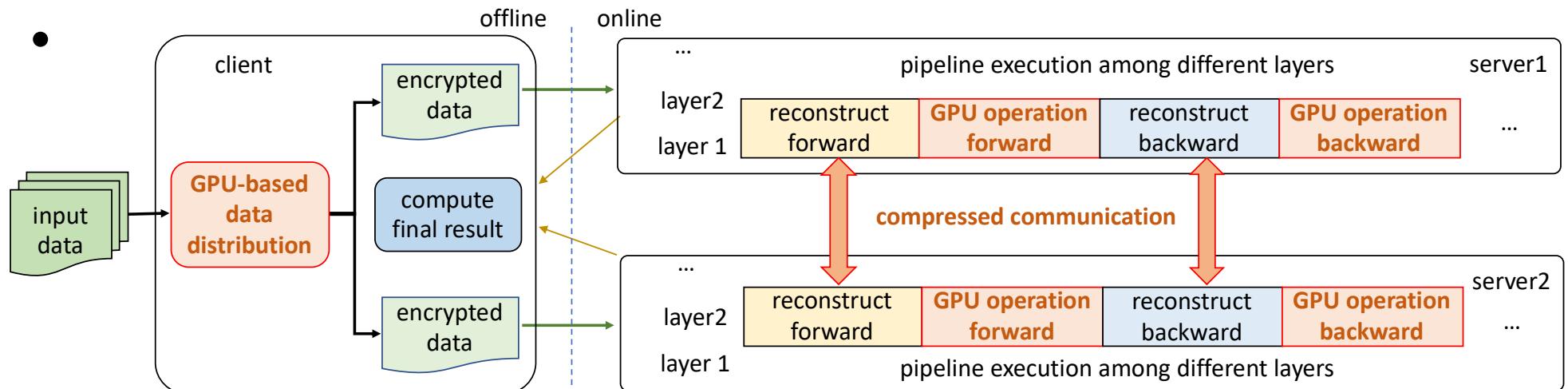
4. Challenges

- Challenge 3: Complicated inter-node data dependence



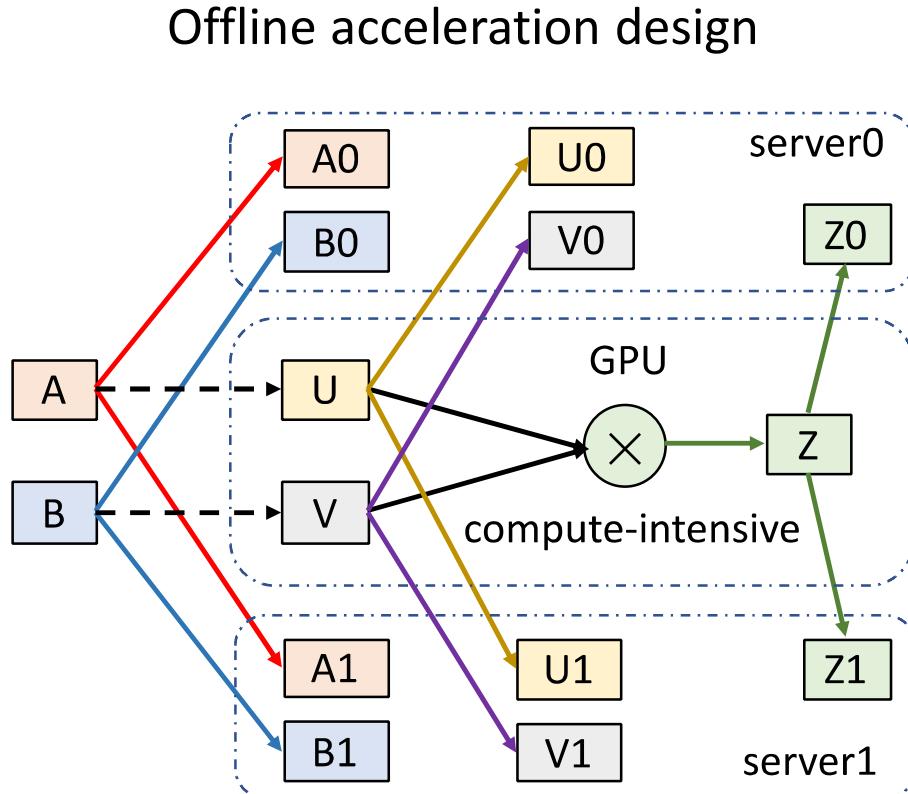
5. ParSecureML

- Overview - ParSecureML consists of three major components:
 - Profiling-guided adaptive GPU utilization
 - Intra-node double pipeline
 - Inter-node compressed transmission communication



5. ParSecureML

- Profiling-Guided Adaptive GPU Utilization



Online acceleration design

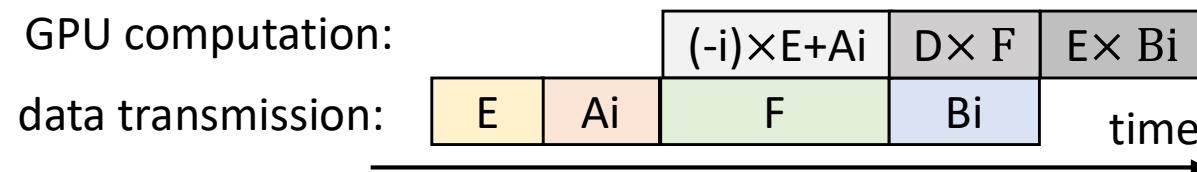
$$CPU: \quad E_i = E_0 + E_1 \quad F = F_0 + F_1$$
$$GPU: \quad C_i = (-i) \times E \times F + A_i \times F + E \times B_i + Z_i$$

$$C_i = \begin{pmatrix} (-i) \times E & A_i & E \end{pmatrix} \times \begin{pmatrix} F \\ F \\ B_i \end{pmatrix} + Z_i, i \in 0, 1$$

$$C_i = \begin{pmatrix} (-i) \times E + A_i & E \end{pmatrix} \times \begin{pmatrix} F \\ B_i \end{pmatrix} + Z_i, i \in 0, 1$$

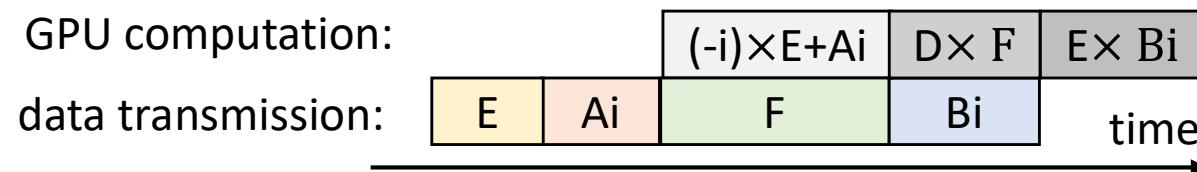
5. ParSecureML

- Double Pipeline for Intra-Node CPU-GPU Fine-Grained Cooperation
 - Pipeline 1 to overlap PCIe data transmission and GPU computation.

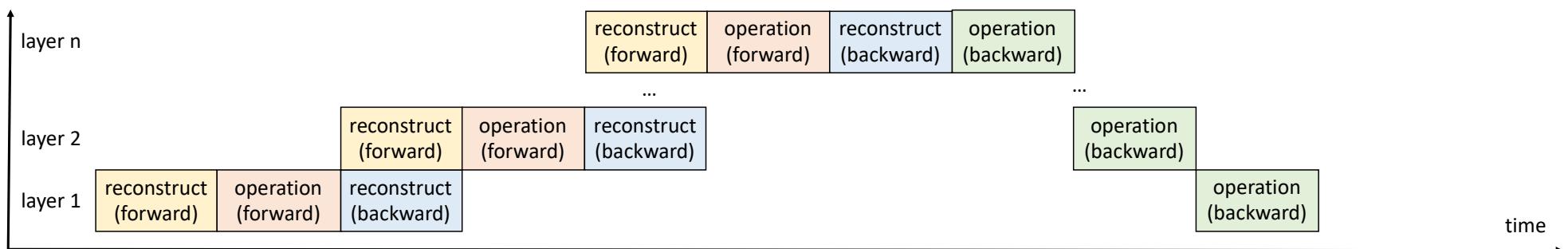


5. ParSecureML

- Double Pipeline for Intra-Node CPU-GPU Fine-Grained Cooperation
 - Pipeline 1 to overlap PCIe data transmission and GPU computation.

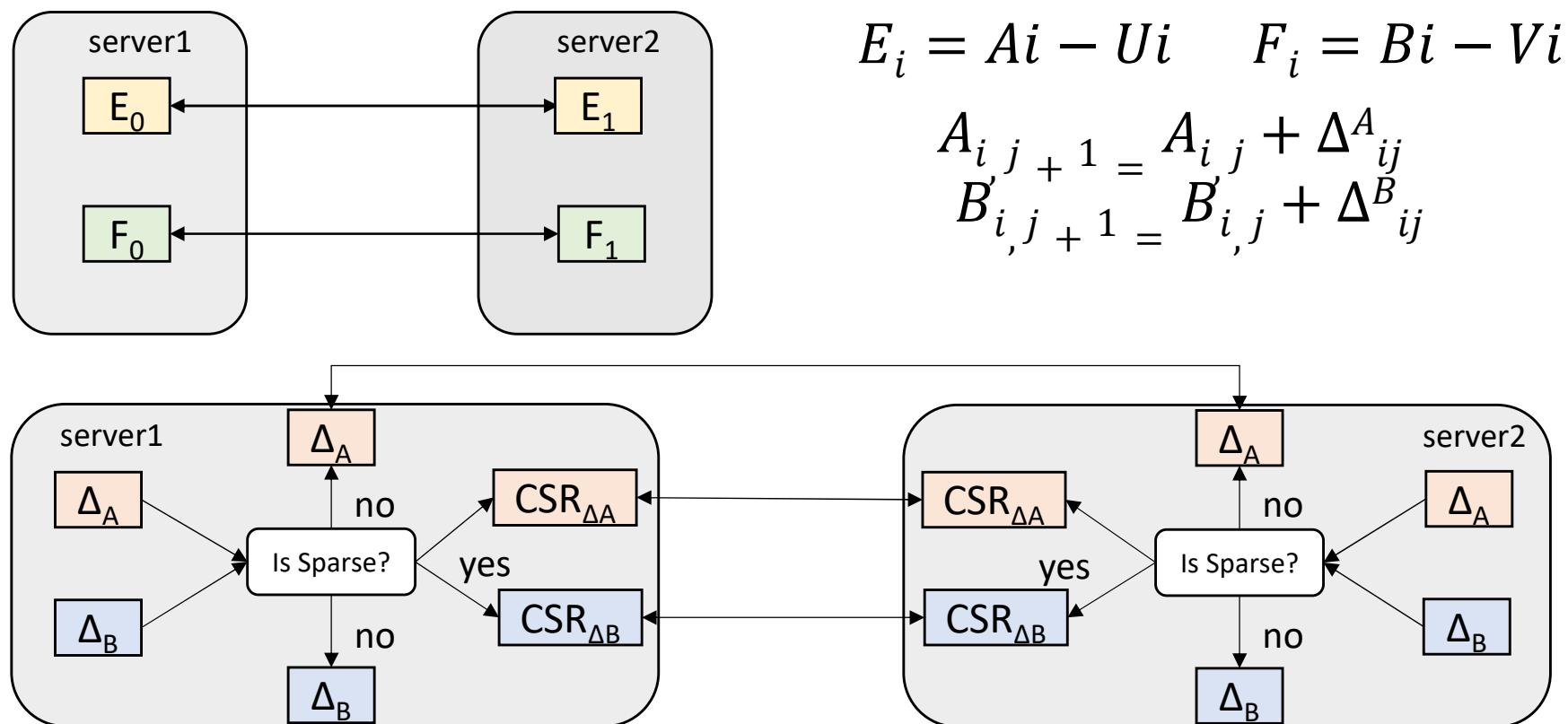


- Pipeline 2 to overlap operations



5. ParSecureML

- Compressed Transmission for Inter-Node Communication



6. Evaluation

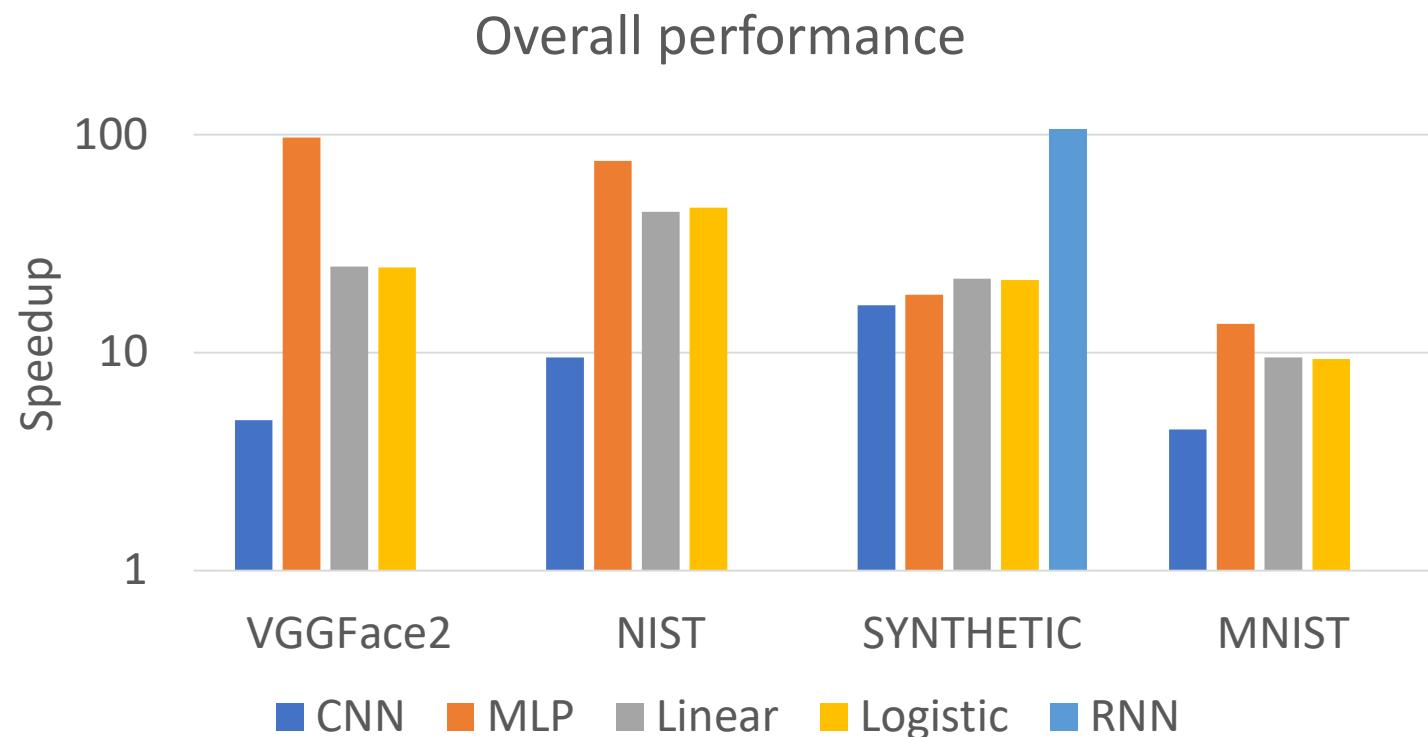
- Baseline: SecureML[1]
- Benchmarks
 - Convolution neural network (CNN)
 - Multilayer Perceptron (MLP).
 - Recurrent neural network (RNN)
 - Linear regression
 - Logistic regression
- Datasets - VGGFace2/NIST/SYNTHETIC/MNIST
- HPC Cluster
 - Intel(R) Xeon(R) CPU E5-2670 v3
 - Nvidia Tesla V100



[1] Mohassel P, Zhang Y. Secureml: A system for scalable privacy preserving machine learning[C]//2017 IEEE Symposium on Security and Privacy (SP). IEEE, 2017: 19-38

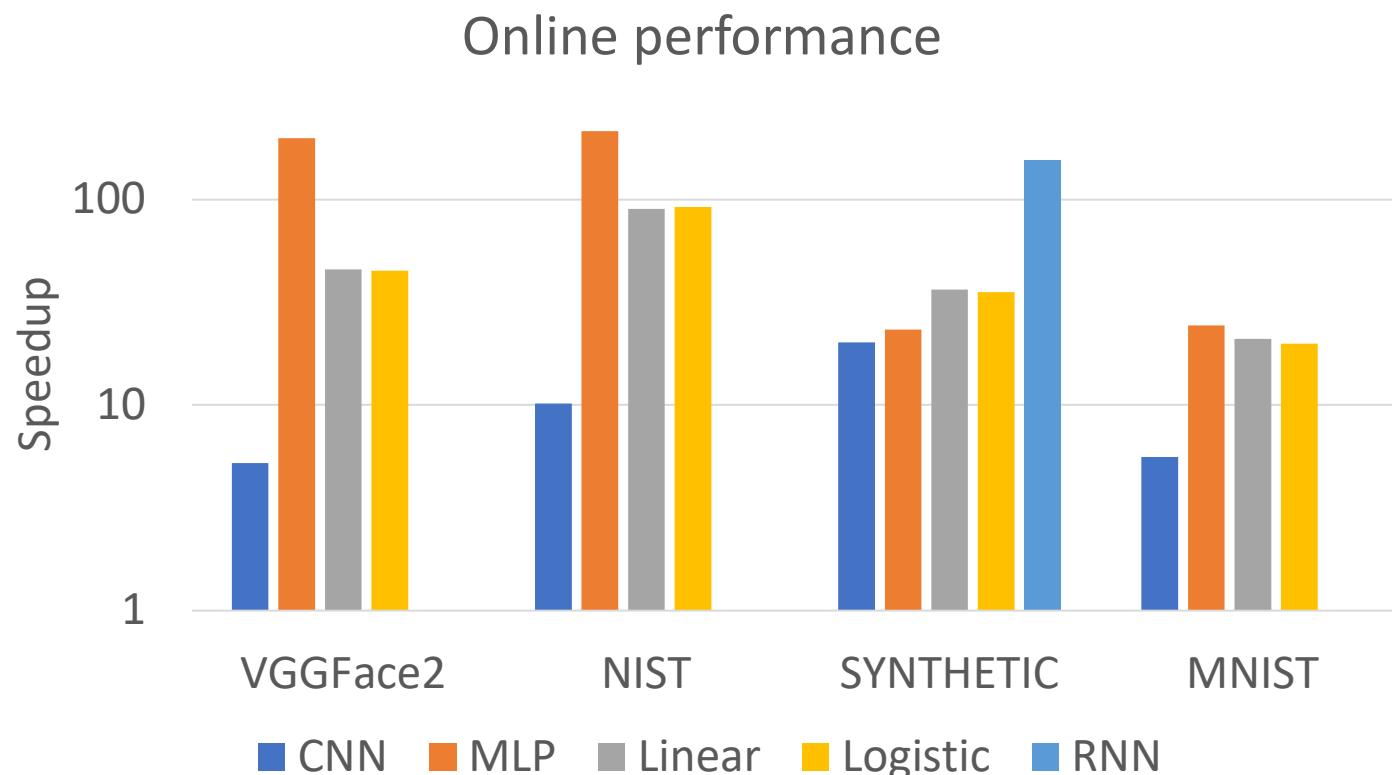
6. Evaluation

- Overall speedups. On average, ParSecureML achieves an average speedup of 32.2x over the SecureML.



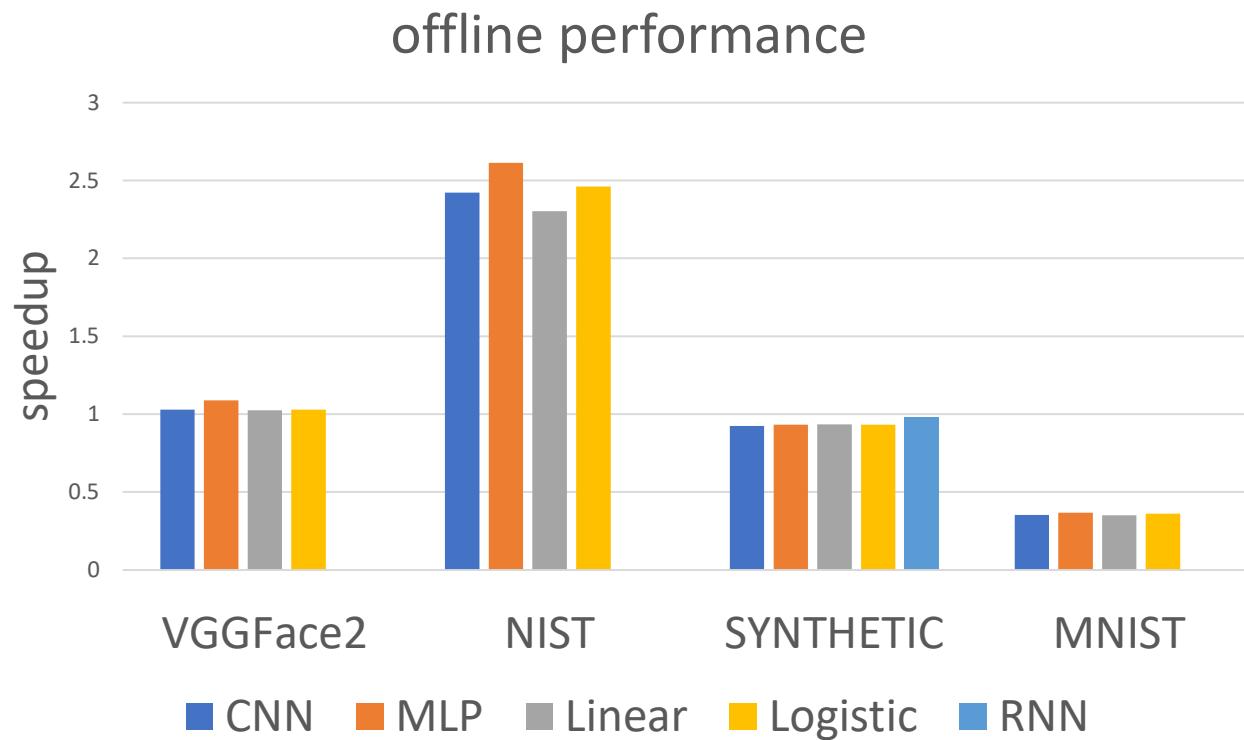
6. Evaluation

- Online speedups. The average online performance speedup is 61.4x (even higher than the overall speedup).



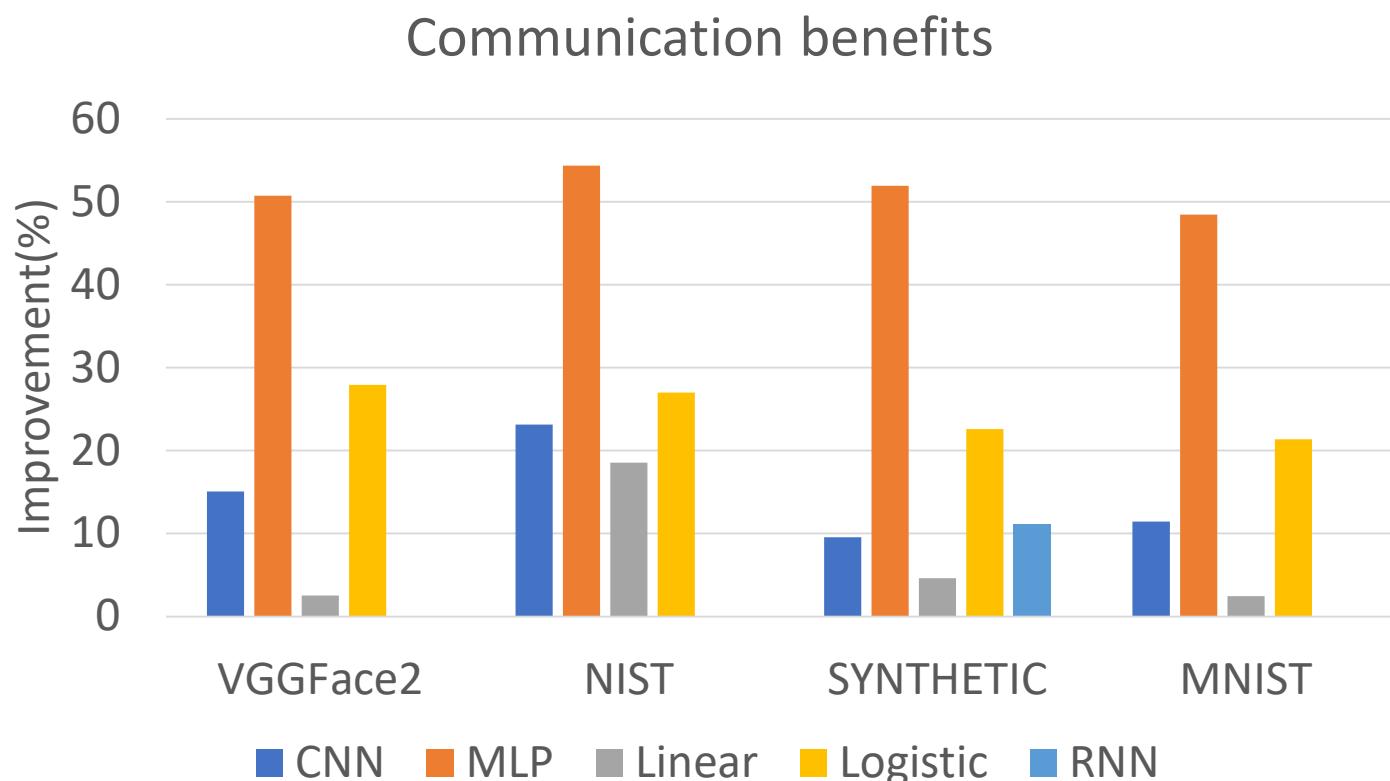
6. Evaluation

- Offline speedups. Applying GPUs in the offline phase brings 1.2x performance benefits.



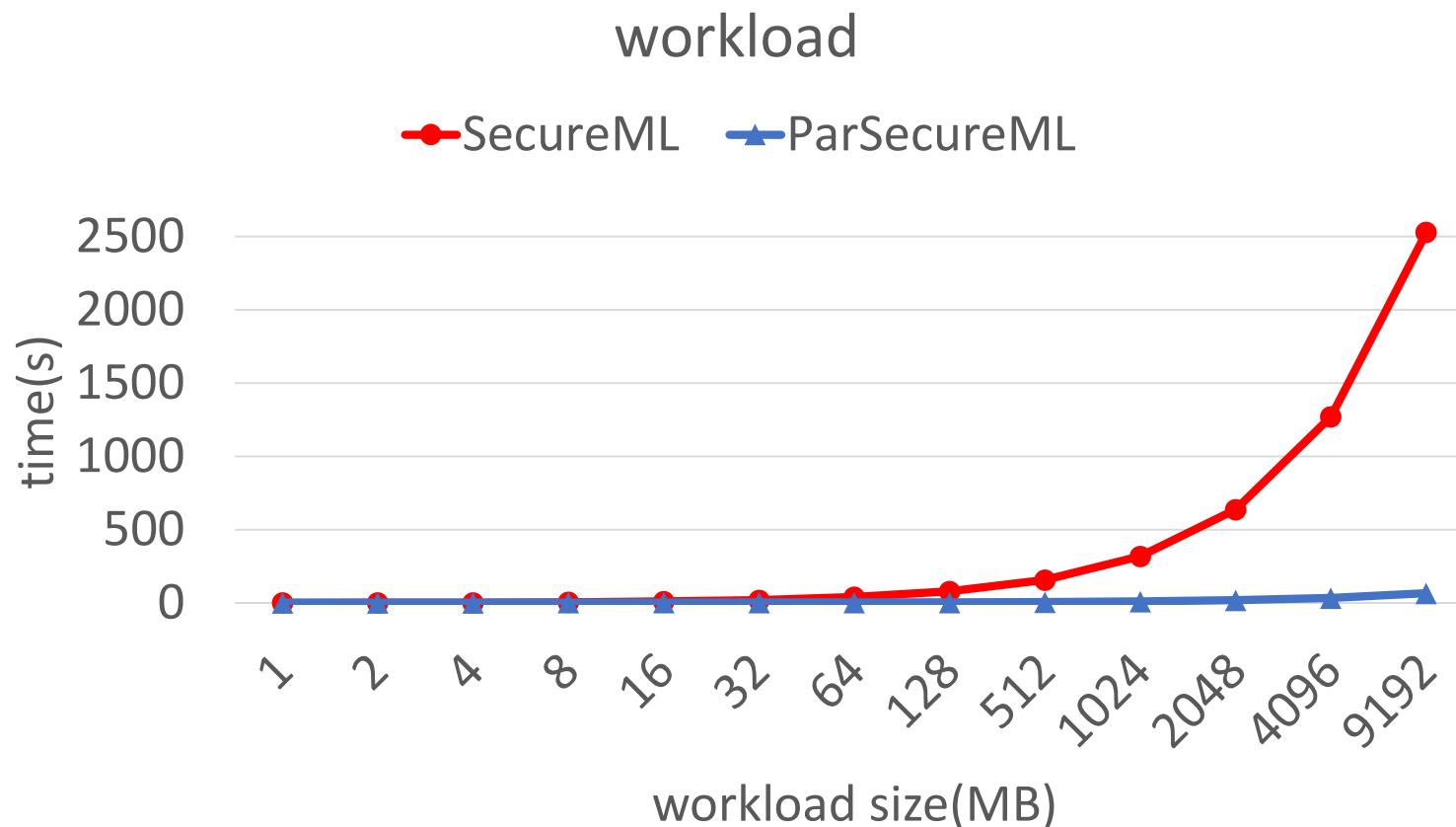
6. Evaluation

- Communication benefits - On average, ParSecureML reduces 23.7% communication overhead.



6. Evaluation

- Influence of workload size



6. Source Code at Github

- <https://github.com/ZhengChenCS/ParSecureML>

The screenshot shows the GitHub repository page for 'ZhengChenCS / ParSecureML'. The page includes a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The main content area displays the 'Code' tab, showing the 'master' branch with 1 branch and 0 tags. A recent commit by 'ZhengChenCS' is shown, merging the 'master' branch from 'https://github.com/ZhengChenCS/ParSecureML'. The commit was made 8 days ago and contains 8 commits. Below the commit log, there is a list of files: bin, build, include, source, test, LICENSE, Makefile, and README.md. The 'About' section describes the repository as a 'Parallel Secure Machine Learning Framework on GPUs'. It includes links for 'Readme' and 'MIT License'. The 'Releases' section indicates 'No releases published' and provides a link to 'Create a new release'. The 'Packages' section indicates 'No packages published' and provides a link to 'Publish your first package'.

ZhengChenCS / ParSecureML

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags Go to file Add file Code

ZhengChenCS Merge branch 'master' of https://github.com/ZhengChenCS/ParSecureML 542ca0b 8 days ago 8 commits

File	Commit Message	Time
bin	a	12 days ago
build	a	12 days ago
include	a	12 days ago
source	a	8 days ago
test	a	12 days ago
LICENSE	Create LICENSE	12 days ago
Makefile	a	12 days ago
README.md	Update README.md	12 days ago

About

A Parallel Secure Machine Learning Framework on GPUs

Readme MIT License

Releases

No releases published Create a new release

Packages

No packages published Publish your first package

7. Conclusion

- We exhibit our observations and insights in SecureML acceleration.
- We develop ParSecureML, the first parallel secure machine learning framework on GPUs.
- We demonstrate the benefits of ParSecureML over the state-of-the-art secure machine learning framework.

Thank you!

- Any questions?

ParSecureML: An Efficient Parallel Secure Machine Learning Framework on GPUs

Zheng Chen[◊], Feng Zhang[◊], Amelie Chi Zhou[★],

Jidong Zhai⁺, Chenyang Zhang[◊], Xiaoyong Du[◊]

[◊]Renmin University of China

[★]ShenZhen University

⁺Tsinghua University

chenzheng123@ruc.edu.cn, fengzhang@ruc.edu.cn, chi.zhou@szu.edu.cn,
zhaijidong@tsinghua.edu.cn, chenyangzhang@ruc.edu.cn, duyong@ruc.edu.cn

