Transforming Data Sharing: Unleashing the power of Intel platforms in federated learning: A comprehensive evaluation

國立臺北護理健康大學
National Taipei University of Nursing and Health Sciences

Associate Professor, Chung-Yueh Lien
Bo-An, Shen
Yu-Xun, Zhan

Summary

ASUS is an OEM passionate about technology and driven by innovation as well as able to provide from various computer portfolio to AI and data center solutions. In addition to its profound expertise in hardware design, ASUS continues to develop high performance computing, cloud service software architectures to enable overall solutions.

With its expertise in AI applications, ASUS has worked with Intel to realize Federated Learning training by using 4th Gen Intel® Xeon® Scalable processor. In this paper, ASUS presents a comprehensive AI solution based on Federated Learning training models and Intel's exclusive AI optimization methods. The purpose of this study is to demonstrate ASUS' integration with CPU computing power and AI technologies. A meticulous investigation has been conducted by ASUS into the experimental design and methodology used to assess the performance of Intel CPUs within the dynamic landscape of Federated Learning (FL). Intel® Xeon® Scalable processors are designed to accelerate performance across a wide range of workloads, including artificial intelligence (AI), data analytics, networking, storage, and high-performance computing. In this field study, ASUS worked with Imaging Lab, National Taipei University of Nursing and Health Sciences, which is the professional research team focusing in smart medical and AI workloads. Dataset processing, AI Model formation, are handled by Shen, Bo-An and Zhan, Yu-Xun.

Agenda

Introduction/background

Federated Learning is a machine learning approach that enables training models across decentralized edge devices (such as smartphones, IoT devices, or edge servers) without exchanging raw data. The main idea behind Federated Learning is to keep the data localized, addressing privacy concerns, and reducing the need to transmit sensitive information to a central server. With Federated Learning which can facilitate the training of machine learning models across a multitude of decentralized devices. Within this paradigm, Federated Learning (FL) extends the collaborative ethos, fostering cooperation within FL experiments. This paper delves into the analytical evaluation of Intel CPU performance within this cutting-edge context.

Challenges in Federated Learning

Federated Learning initiatives are pivotal in propelling the field of medical AI forward. They achieve this by offering funding, setting standards, regulating practices, encouraging collaboration, and supporting education and workforce development. Through the utilization of these initiatives, stakeholders within the healthcare ecosystem can tap into the transformative capabilities of AI, thereby enhancing patient outcomes, refining clinical decision-making processes, and spurring innovation in medical research and practice. However, addressing critical issues such as AI architectures, data management, and accuracy across multiple locations remains paramount. Optimizing algorithms, hardware platforms, and methodologies are pivotal factors that can amplify the efficacy of Federated Learning implementations in medical segments. It is imperative to strategize and invest in solutions that streamline these technical aspects, ensuring seamless integration and robust performance of AI-driven systems across diverse healthcare settings. By prioritizing these considerations, Federated Learning initiatives can further bolster their impact and facilitate the broader adoption of AI technologies in healthcare.

Interagency Coordination

Coordinating AI efforts across different federated agencies with varying priorities, budgets, and organizational structures can be challenging. Aligning strategies, sharing resources, and fostering collaboration among agencies require effective communication, leadership, and governance mechanisms.

Data Access and Quality

Accessing high-quality data for training AI models while preserving patient privacy and confidentiality is a major hurdle. Federated Learning approaches, data sharing agreements, and data anonymization techniques can help mitigate data access challenges, but ensuring data quality and integrity remains an ongoing concern.

Technical Infrastructure and Expertise

Building and maintaining the technical infrastructure required for AI development, deployment, and operation is resource-intensive. Investing in hardware, software, cloud computing, and cybersecurity capabilities, as well as recruiting and retaining skilled AI professionals, are key priorities for federated agencies seeking to advance their AI initiatives.

How to Enable a Comprehensive AI Platform

To speed up Federated Learning adoption in the smart medical industry, the ASUS data center solution team is trying to adopt the latest Intel® Xeon® Scalable processors to optimize Federated Learning performance with Intel built-in AI software toolkits and solutions. With this cutting-edge technology, 5th Gen Intel® Xeon® Scalable Processors delivers better performance to improve overall AI efficiency and accuracy.

Why ASUS adopt Intel platform in Federated Learning?

Built-in AI technology

The 4th Gen Intel® Xeon® Scalable Processors stand out as a groundbreaking solution, embedding AI acceleration directly into each core. This innovative design streamlines the processing of AI workloads without relying on additional discrete accelerators. As a result, Intel's platform emerges as the pinnacle of performance, striking a harmonious balance between AI tasks and other workloads. Specifically tailored to meet the rigorous demands of Federated Learning applications, Intel's platform boasts specialized features and optimizations. Its robust security protocols and scalable architecture render it a dependable choice for managing sensitive data and addressing the intricate computational requirements of government agencies. Notably, Intel's processors deliver up to a remarkable 42% enhancement in inference performance while achieving latency of fewer than 100 milliseconds for large language models (LLMs) containing up to 20 billion parameters. This innate AI capability translates into expedited and more efficient execution of AI tasks, underscoring Intel's commitment to empowering advanced computing capabilities tailored for the demands of Federated Learning initiatives.

Optimization in AI

Intel offers a suite of optimization tools designed to enhance performance and efficiency in various machine-learning workflows. These solutions are tailored to address specific challenges encountered in modern computing environments, empowering developers, and data scientists to achieve superior results across a range of tasks. Key components of Intel's optimization toolkit include:

- Intel® Distribution for Python

The Intel Distribution for Python provides a high-performance Python environment optimized for Intel processors. It includes popular libraries such as NumPy, SciPy, and scikit-learn, all of which are optimized to leverage the advanced capabilities of Intel architecture. This distribution enables accelerated execution of Python-based machine learning and data analysis tasks, leading to significant improvements in performance and efficiency.

- Installation:
  Detailed instructions for installing Intel Distribution for Python Installation are available in the Intel Distribution for Python Installation documentation.
- conda install -c intel -c conda-forge --override-channels modin-all python=3.10

- Intel® Extension for PyTorch

The Intel Extension for PyTorch extends the capabilities of the popular PyTorch deep learning framework by integrating optimizations specifically tailored for Intel hardware. By leveraging features such as vectorization, threading, and memory optimizations, this extension enables PyTorch-based models to achieve higher levels of performance on Intel processors. It seamlessly integrates with existing PyTorch workflows, allowing users to unlock the full potential of Intel architecture without compromising on productivity.

- Installation:
  Detailed instructions for installing Intel Extension for PyTorch Installation are available in the Intel Extension for PyTorch Installation documentation.
- python -m pip install torch==2.1.2 torchvision==0.16.2 torchaudio==2.1.2 --index-url https://download.pytorch.org/whl/cpu
- python -m pip install intel-extension-for-pytorch==2.1.100
- python -m pip install oneccl_bind_pt==2.1.0 --extra-index-url https://pytorch-extension.intel.com/release-whl/stable/cpu/us/
- python -c "import torch; import intel_extension_for_pytorch as ipex; print(torch.__version__); print(ipex.__version__);" # installation check

  To use Intel Extension for PyTorch, we need to add the optimizer before training.

  model, optimizer= ipex.optimize(model, optimizer=optimizer)

- Intel® oneAPI Deep Neural Network Library (oneDNN)

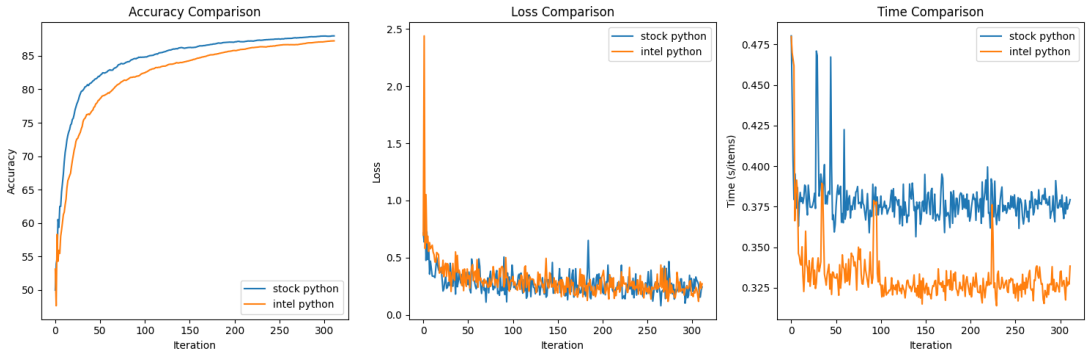
oneDNN (oneAPI Deep Neural Network Library), is a highly optimized deep

learning library. It provides a set of primitives for accelerating deep neural network computations on Intel processors, enabling faster training and inference for a wide range of models. oneDNN is designed to seamlessly integrate with popular deep learning frameworks such as TensorFlow, PyTorch, and MXNet, making it easy for developers to leverage Intel's optimizations without modifying their existing codebase. The oneDNN library has been statically linked in either stock versioned Pytorch or Intel extension for Pytorch CPU. Therefore, there is no need for the extra installation required.

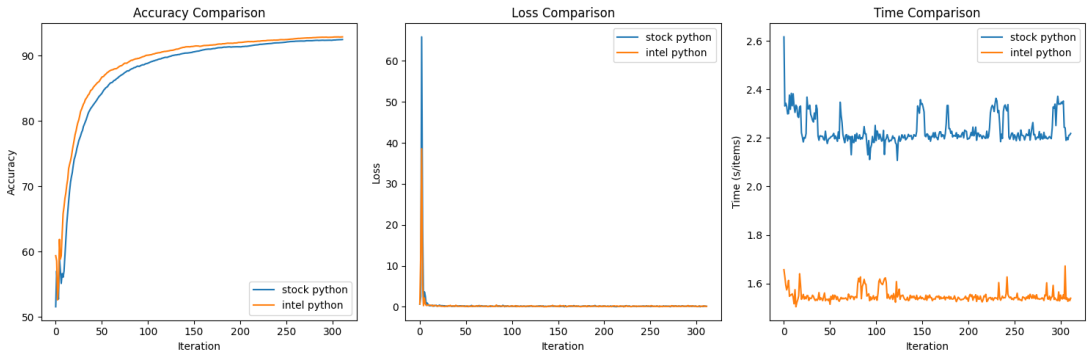
Our Experiment

We tested under settings in the Federated Learning framework(Flower, https://github.com/adap/flower), with Intel solution, from accuracy, loss comparison, and time comparison, which showed the latest Intel solution is the optimized solution to Federated Learning AI.
In this experiment, we aim to investigate the efficacy of leveraging Intel's suite of optimization tools, including oneDNN, Intel Distribution for Python, and Intel Extension for PyTorch, in the context of Federated Learning through Intel® VTune Profiler, a performance profiling tool used to investigate performance issues and learn the performance insights. By conducting controlled experiments using our select AlexNet and VGG19 models, which are commonly evaluated for medical image recognition applications, and the fundamental models for deriving our own research model on the particular medical application, we seek to evaluate the impact of integrating Intel's tools into FL client environments. Our research model's parameter size was estimated to be between those of the AlexNet and VGG19 models. Specifically, our experimental setup includes a control group with standard Python environments and an experimental group where Intel tools are incorporated, allowing for a comparative analysis of training efficiency, convergence rates, and model accuracy. This investigation aims to provide valuable insights into the potential benefits of utilizing Intel's optimization solutions for Federated Learning scenarios.
The experiment code and steps are in this Github repository we provide.

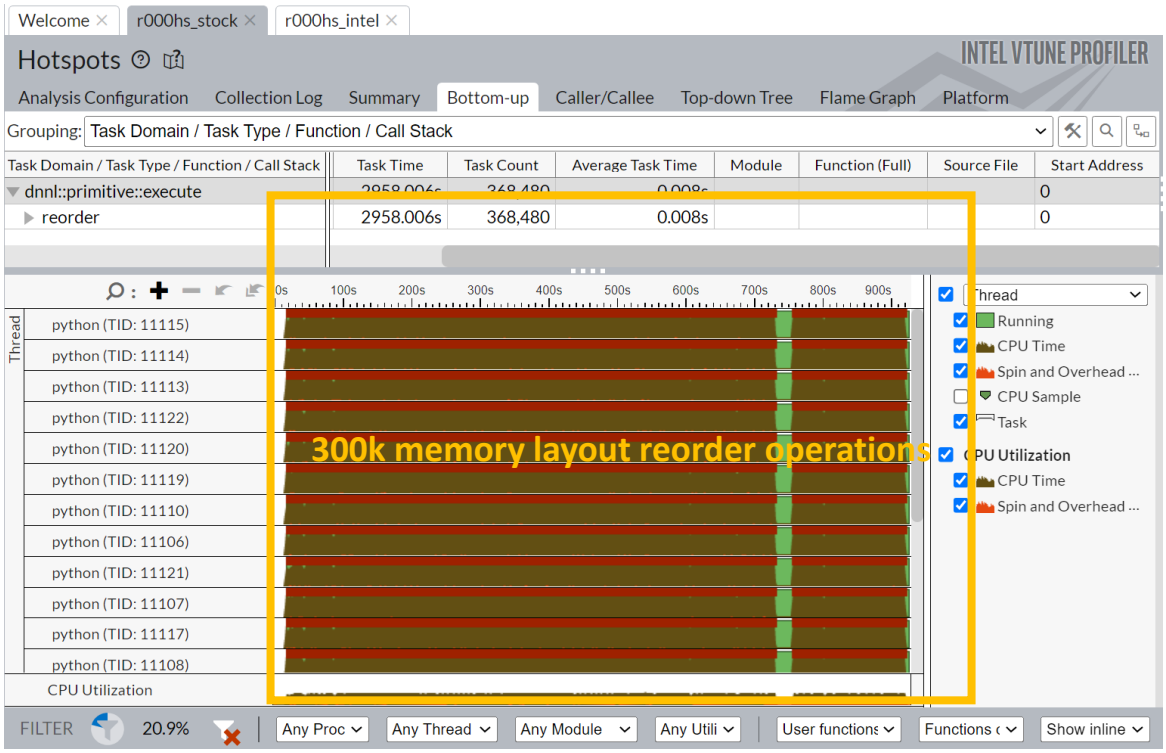
- AlexNet



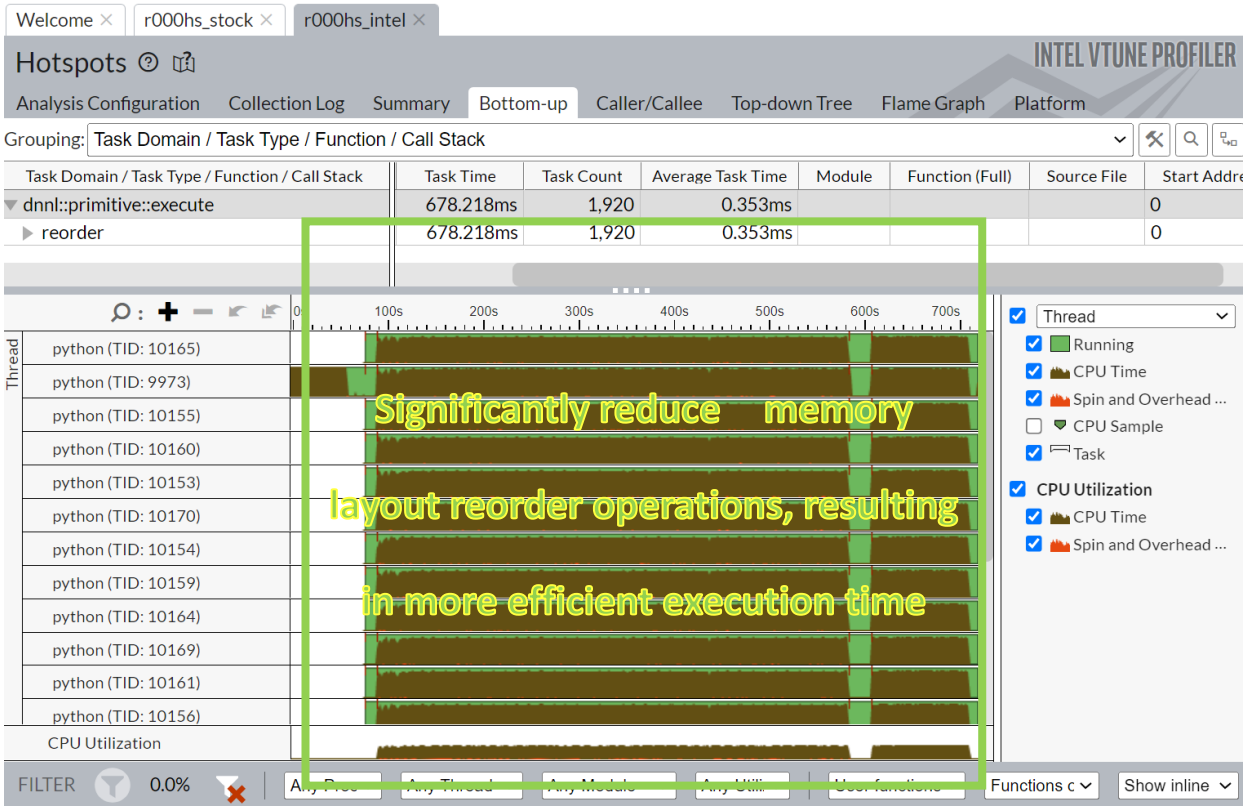
- VGG19



- Intel® VTune™ Profiler results comparison.



VTune Figure 1, oneDNN reorder CPU time consumtion(unoptimized)



VTune Figure 2, oneDNN reorder CPU time consumption (Intel optimization)

Performance (Results & Benefits)

Based on the results of our experiment, it is evident that integrating Intel's optimization tools, including oneDNN, Intel Distribution for Python, and Intel Extension for PyTorch, into federated learning environments can yield substantial improvements in efficiency, particularly for more complex models like VGG19. Our findings reveal an average efficiency improvement of 13% for AlexNet and an impressive 30% for VGG19, indicating that the complexity of the model plays a significant role in the extent of enhancement achievable. Moreover, detailed analysis using Intel VTune uncovered that the primary driver behind the efficiency gains in VGG19 is the reduction of oneDNN's reordering operations, resulting in a remarkable reduction in CPU time from 2958s to 0.6s. While these advancements are promising, it is important to note a crucial limitation: the necessity for all clients in federated learning to adopt Intel Extension for PyTorch if any one client chooses to use it. This requirement introduces potential compatibility constraints and operational complexities within federated learning setups. In conclusion, while our experiment showcases the considerable benefits of leveraging Intel's

optimization tools for federated learning, further exploration is warranted to address the associated challenges and maximize the scalability and usability of such solutions in distributed machine learning environments.

We have achieved impressive efficiency gains by collaborating with Intel and National Taipei University of Nursing and Health Sciences (NTUNHS) on implementing Federal Learning medical image recognition in ASUS's server products with Intel 4th Gen Xeon Scale Processors. Integrating Intel oneAPI software tools and Intel Extension for PyTorch into Federated Learning environments has resulted in a 13% increase in efficiency for AlexNet and a 30% increase for VGG19 in medical image recognition applications. With the help of Intel oneAPI's VTune, we found that OneDNN's operation reordering significantly reduced CPU computation time for deep learning model training. Intel Software optimization tools deliver values by enhancing performance in distributed machine learning environments without compromising data privacy."

- Paul Ju, Corp VP, CTO & Infrastructure Solution Group GM, ASUS

Federated Learning Test Result - Training Accuracy

The comparison of train accuracy, which tracks the cumulative number of correct predictions during training, between two configurations of Python—the stock community version and the Intel optimization version (Intel Distribution for Python and Intel Extension for Pytorch) — in the context of Federated Learning tasks. It shows how well the model is learning from data Despite introducing randomness in dataset preprocessing, both configurations exhibit similar stability across different training rounds, ultimately resulting in a converged model.

Federated Learning Test Result - Train Loss

In the domain of loss measurement, the stock version of Python showcases marginally reduced values in comparison to the Intel optimization version, which registers a slight uptick of approximately 0.1 percentage points. This discrepancy, although seemingly minor, holds significant implications for the optimization of Federated Learning (FL) scenarios. The observation underscores the critical importance of honing Intel's optimization strategies to ensure optimal performance in FL environments. Given the intricate nature of FL tasks, even subtle deviations in loss metrics can signify underlying

nuances in algorithmic execution and model convergence. Consequently, Intel's optimization methodologies become imperative for achieving precision and efficiency in FL operations. By addressing these nuanced discrepancies and continually refining optimization strategies, we can bolster the Federated Learning framework's capacity to meet the evolving demands of FL applications, thereby advancing the landscape of distributed machine learning paradigms.

Federated Learning Test Result – Training Time

In terms of time efficiency, a standout discovery emerges regarding the computational performance of the two configurations. The Intel optimization Python development environment notably surpasses the stock version Python environment concerning processing time, boasting a nearly 30% performance advantage in the average iteration training time in this test. This considerable decrease in processing time underscores the tangible advantages of harnessing Intel's CPU optimizations, particularly in Federated Learning (FL) environments where efficiency holds paramount importance.

Competitive Advantage

In summary, while the stock version of Python may demonstrate a marginal advantage in accuracy and loss metrics, the Intel Distribution for Python presents a compelling superiority in computational efficiency, evidenced by markedly faster processing times. These discoveries underscore the potential of Intel's CPU optimization techniques to augment the performance of Federated Learning (FL) frameworks, especially in demanding healthcare applications where real-time processing and efficiency are imperative. Nevertheless, continued optimization efforts are needed to narrow the accuracy gap observed between the two configurations. By addressing this disparity, Intel can further solidify its position as a leading provider of optimized computing solutions for FL applications, facilitating the advancement of healthcare analytics and patient-centric care delivery. Through collaborative endeavors and ongoing innovation, stakeholders can harness the full potential of Intel's technology advancements to propel the evolution of FL frameworks, ultimately enhancing the quality and efficiency of healthcare services on a global scale.

Conclusion

In conclusion, our experiment has shed light on the remarkable potential of integrating Intel's optimization tools into federated learning environments, unlocking significant improvements in efficiency for complex models like VGG19. With an average efficiency gain of 13% for AlexNet and a staggering 30% for VGG19, our findings demonstrate the transformative power of Intel's optimization tools in accelerating the performance of federated learning. As we delve deeper into the intricacies of our results, it's clear that the reduction of oneDNN's reordering operations holds the key to these remarkable gains, slashing CPU time from a staggering 2958s to a mere 0.6s. While these advancements are undeniably promising, we must acknowledge the crucial limitation that requires all clients to adopt Intel Extension for PyTorch to reap the benefits. As we continue to push the boundaries of federated learning, we're excited to explore the vast possibilities that these optimizations hold, and we invite you to join us on this journey of innovation by learning more about Federated AI.

Please stay tuned an upcoming thorough assessment of ASUS's Intel® Data Center GPU Flex Series server is set to reveal more impressive outcomes, demonstrating the server's effectiveness in assisting actual medical disease cases.

Learn More

- Asus Federated AI
- Intel Ecosystem Developer Resources Hub Test Configurations

FL Server Configuration

- Server System: ASUS RS720Q-E11-RS8U
- Operating System: Debian 12
- CPU: - Intel® Xeon® Silver 4410T Processor (10 cores per socket, Hyper-Threading ON / 2.7Ghz Base Frequency / 150W TDP) x 2
- RAM: 128 GB 4x32GB DDR5 4800 MT/s [4000 MT/s] Samsung M321R4GA0BB0-CQKEG

- Storage: SAMSUNG MZ7L31T9, 1.7TB
- Network: Ethernet Controller X710 for 10GBASE-T

FL Client Configuration

- Server System: ASUS RS720Q-E11-RS8U
- Operating System: Debian 12
- CPU: - Intel® Xeon® Silver 4410T Processor (10 cores per socket, Hyper-Threading ON / 2.7Ghz Base Frequency / 150W TDP) x 2
- RAM: 128 GB 4x32GB DDR5 4800 MT/s [4000 MT/s] Samsung M321R4GA0BB0-CQKEG
- Storage: SAMSUNG MZ7L31T9, 1.7TB
- Network: Ethernet Controller X710 for 10GBASE-T

Experiment Group Setup

Control Group

- Server: Python3.10, no addition modification
- Client1: Python3.10, no addition modification
- Client2: Python3.10, no addition modification

Experimental Group

- Server: Python3.10, no addition modification
- Client1: oneAPI base toolkit 2024.0 installed, Intel Distribution for Python 3.10 installed, Intel Extension for PyTorch cpu v2.1.1 implemented.
- Client2: oneAPI base toolkit 2024.0 installed, Intel Distribution for Python 3.10 installed, Intel Extension for PyTorch cpu v2.1.1 implemented.

Contact Us

Customize your solutions by leaving ASUS a message. Whether you need a server sample, solution consultation, design assistance or have other

questions, we're here to help!
Learn more about ASUS server: https://servers.asus.com/
Customize your solution today: https://servers.asus.com/support/contact
Access to the supplementary material: https://github.com/cylab-tw