



HAL
open science

Towards Efficient Learning on the Computing Continuum: Advancing Dynamic Adaptation of Federated Learning

Mathis Valli, Alexandru Costan, Cédric Tedeschi, Loïc Cudennec

► To cite this version:

Mathis Valli, Alexandru Costan, Cédric Tedeschi, Loïc Cudennec. Towards Efficient Learning on the Computing Continuum: Advancing Dynamic Adaptation of Federated Learning. FlexScience 2024 - 14th Workshop on AI and Scientific Computing at Scale using Flexible Computing Infrastructures, Jun 2024, Pisa, Italy. pp.42-49, 10.1145/3659995.3660042 . hal-04698619v2

HAL Id: hal-04698619

<https://hal.science/hal-04698619v2>

Submitted on 16 Sep 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Towards Efficient Learning on the Computing Continuum: Advancing Dynamic Adaptation of Federated Learning

Mathis Valli, Alexandru Costan, Cédric Tedeschi
University of Rennes, Inria, CNRS, IRISA
Rennes, France
{mathis.valli,alexandru.costan,cedric.tedeschi}@irisa.fr

Loïc Cudennec
DGA Maîtrise de l'Information
Rennes, France
loic.cudennec@intradef.gouv.fr

ABSTRACT

Federated Learning (FL) has emerged as a paradigm shift enabling heterogeneous clients and devices to collaborate on training a shared global model while preserving the privacy of their local data. However, a common yet impractical assumption in existing FL approaches is that the deployment environment is static, which is rarely true in heterogeneous and highly-volatile environments like the Edge-Cloud Continuum, where FL is typically executed. While most of the current FL approaches process data in an online fashion, and are therefore adaptive by nature, they only support adaptation at the ML/DL level (e.g., through continual learning to tackle data and concept drift), putting aside the effects of system variance. Moreover, the study and validation of FL approaches strongly rely on simulations, which, although informative, tends to overlook the real-world complexities and dynamics of actual deployments, in particular with respect to changing network conditions, varying client resources, and security threats.

In this paper we make a first step to address these challenges. We investigate the shortcomings of traditional, static FL models and identify areas of adaptation to tackle real-life deployment challenges. We devise a set of design principles for FL systems that can smartly adjust their strategies for aggregation, communication, privacy, and security in response to changing system conditions. To illustrate the benefits envisioned by these strategies, we present the results of a set of initial experiments on a 25-node testbed. The experiments, which vary both the number of participating clients and the network conditions, show how existing FL systems are strongly affected by changes in their operational environment. Based on these insights, we propose a set of take-aways for the FL community, towards further research into FL systems that are not only accurate and scalable but also able to dynamically adapt to the real-world deployment unpredictability.

KEYWORDS

federated learning, dynamic adaptation, computing continuum, machine learning, data privacy

1 INTRODUCTION

Federated Learning (FL) has emerged as a distributed learning paradigm that allows multiple clients to collaboratively address learning tasks without local data sharing [25]. Unlike traditional centralized machine learning (ML) models, which require the aggregation of extensive amounts of sensitive data on a single server, often raising data privacy and security concerns, FL adopts a fundamentally different approach. It enables ML models training locally on individual devices or servers, with only model updates shared, rather than raw data. We focus on the cross-device case, where this global learning

task is typically executed across the Edge-Cloud (or the Computing Continuum [1]): a local model is trained on each client (at the Edge, close to the data sources) and only its model parameters are shared with a central server (in the Cloud, or HPC center). The central server handles the model initialization, the selection of participating clients for each federated round, the aggregation of all received parameters into global model parameters, and the distribution of the updated global parameters back to the clients. This process is illustrated in Figure 1.

Practical FL deployments across such Edge-Cloud Continuum, typically leveraging geographically-distributed and heterogeneous resources, face several challenges at both *system* and *data* levels. At *system* level, the network or the computing resources may become suddenly limited and the number of active clients can fluctuate significantly. At *data* level, the data characteristics may evolve unexpectedly, as in the case of DL applications that increasingly need to be trained with unbounded datasets that are updated frequently. For instance, scientific applications using experimental devices such as sensors need to quickly analyze the experimental data in order to steer an ongoing experiment. These fluctuations can impact the learning process through increased resource consumption (network, GPU/CPU hours, storage space), leading to both prohibitive runtimes as well as inefficient resource usage.

Recently, a growing body of research has focused on addressing the latter, data variance issues. Heterogeneous data among clients is handled through Personalized Federated Learning [34], in which the federated system not only aggregates global model updates from participating devices but also allows for personalized model customization. Since local updates to client models may deviate to varying degrees, potentially leading to suboptimal outcomes as the average global model drifts away [8], several solutions focus on Clustered Federated Learning [31], employing multi-task learning [33] to mitigate data heterogeneity by grouping clients into clusters with higher internal similarity (more homogeneity). While these training strategies may provide effective models for coping with evolving data, they overlook the system variance as they typically assume a level of environmental stability that does not hold up in real-world scenarios. Hence, they cannot guarantee the same level of performance when the underlying system conditions vary. This is especially problematic for FL deployments across the Computing Continuum, where Edge devices have a high volatility while Cloud resources are also variable (e.g., due to multi-tenancy).

In this paper, we are focusing precisely on the impact of system variance on the FL performance (i.e., accuracy and training time). We study the performance degradation of existing solutions when deployed on real-clusters. We show that existing schemes fail to

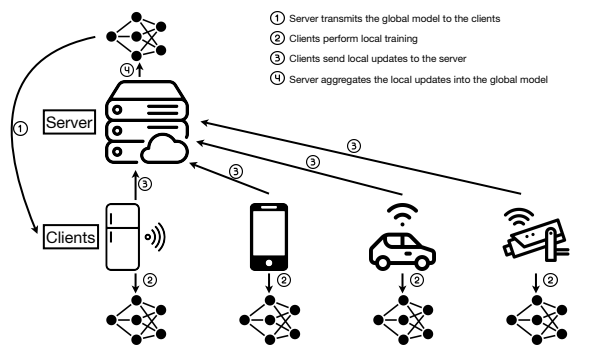


Figure 1: Steps of a Federated Learning process.

preserve accuracy and degrade system efficiency when the environment characteristics change over time. To mitigate this issue and accelerate model convergence, we advocate for dynamic adaptation within FL frameworks. Instead of introducing new techniques, we show how the existing methods can be extended to better cope with the system variability across the Computing Continuum.

Our main contributions are summarized as follows:

- A study of **how different types of system changes affect FL**, highlighting *what*, *when* and *how* to adapt.
- A review of **existing methods that can be adjusted or repurposed** to meet specific challenges such as network inconsistencies, varying number of clients, and evolving data sets.
- An **experimental evaluation on 25 nodes from 2 distant sites of the Grid’5000 [5] testbed**, illustrating the impact of the available bandwidth, latency and client volatility on the training time. The results show that existing FL approaches (i.e., FedAvg [25]) are not designed to handle dynamic adaptation.
- A set of **take-aways for the FL community** on how to incorporate dynamic adaptation in FL frameworks and open related research challenges.

The remainder of this paper is organized as follows. Section 2 reviews the recent body of work addressing the challenges of data heterogeneity in FL and advocates for new research around dynamic adaptation to tackle system variance. Section 3 highlights the envisioned benefits of dynamically adapting to system variance in FL. Section 4 presents a set of design principles for enhancing FL with dynamic adaptation. Section 5 presents the results of our real-world experiments showing how existing FL approaches are impacted by system variance. Finally, Section 6 provides a set of hints for the community on how to elaborate on early adaptive FL strategies.

2 BACKGROUND AND RELATED WORK

2.1 Federated Learning

Federated Learning is an approach to machine learning devised to train models across decentralized and distributed data sources while preserving data privacy and reducing communication overhead. It tackles the difficulty of training AI models on sensitive or private

data without centralizing that data in one location. Among the numerous proposed FL algorithms [17, 36], Federated Averaging, commonly known as FedAvg [25], stands out as a fundamental and extensively utilized algorithm. In FedAvg, a global model is derived by averaging the models from clients at a server and then disseminating the averaged model back to the clients after each training round.

Nevertheless, the canonical FL setup faces several challenges, which can be largely divided into two categories: (1) data heterogeneity (variations in data distribution), and (2) device heterogeneity (disparities in computational capacity, network connection, etc.). The vast majority of FL research is focusing on addressing the first (data) challenge, which complicates the learning of a single shared global model for all clients. We recall the main solutions to these issues in Section 2.2. We discuss the device heterogeneity and how this issue is worsened by the inherent variability of resources and clients on the Computing Continuum in Section 2.3.

2.2 Data Heterogeneity Challenges and Solutions

Data Diversity and Non-IID Data. Data heterogeneity refers to the uneven distribution of features, labels, and/or samples among client datasets. This non-uniformity results in gradient drift between local updates and server updates (which occur post-aggregation). Gradient drift can occur from various sources, including the client model’s overfitting locally biased data, biases in aggregation, and covariate shift. Consequently, gradient drift introduces noise into the training process of the global model and adversely affects its overall performance [20]. To tackle this issue, a new branch of FL referred to as Personalized Federated Learning [34] has emerged, leveraging personalized models for each client. Recently, numerous techniques have been introduced to enhance the accuracy and convergence of personalized FL. These advancements encompass strategies such as clustering [31], multi-task learning [33], meta-learning [11], model mixture [14], or adjusting the influence of updates from different clients based on their data characteristics [7] or contribution quality [37].

Data Transfers Overhead. FL requires regular updates between clients and the central server, which can be problematic in environments with limited bandwidth or high latency, hindering FL’s efficiency and scalability [18]. To cut down on communication overhead, there are strategies for selectively engaging clients in the training phase, targeting clients with either more relevant data or better network connection [12].

Slow Model Learning. The distributed nature of FL, coupled with variations in data often results in slower convergence rates for the global model. This requires additional training rounds to achieve satisfactory performance, impacting the timely deployment of updated models [22]. Approaches based on the dynamic management of local epochs aim to address this issue. These strategies find a middle ground between the local processing done by clients and the need for global communication, to speed up the model learning by adjusting how often updates occur [29].

2.3 System Variance Challenges on the Computing Continuum

Emerging FL applications typically need to be implemented as complex workflows and require the coordinated use of supercomputers, Cloud data centres and Edge-processing devices. This assembly is called the Computing Continuum. It raises challenges at multiple levels: (i) at the application/workflow level, to bridge machine learning and data-driven analytics; (ii) at the middleware level, adequate tools must enable efficient deployment and orchestration of the workflow components across the whole distributed infrastructure; and, finally, (iii) a capable resource management system must allocate a suitable set of components of the infrastructure to run the application workflow, preferably in a dynamic and adaptive way, taking into account the specific capabilities of each component of the underlying heterogeneous infrastructure. In this paper, we are focusing on the FL issues related to the dynamic nature of the Computing Continuum: variable system resources and high client volatility.

Static Settings. Many FL strategies are built on static assumptions regarding client participation, data distribution, and network conditions. Works such as [4, 6] conduct experiments under conditions of constant data distribution and unvarying client engagement. This limits their applicability in dynamic environments where these factors can fluctuate unpredictably.

Isolated Parameters. Several solutions focus on optimizing individual components of FL, such as improving communication efficiency or refining data aggregation techniques, without considering how these aspects are interlinked and change over time. This approach may miss opportunities for broader improvements across FL [21]. As pointed out in [23], much of the research aimed at enhancing accuracy concentrates on optimizing the aggregation algorithm without necessarily considering its dynamic interplay with other FL aspects such as client data heterogeneity or system heterogeneity.

Simulation vs. Experimental Validation. Since the main focus of the FL community is on algorithmic improvements, addressing data and model heterogeneity, the *de facto* standard for validation and evaluation is by simulation. While this approach allows to assess the accuracy of the proposed solutions, it raises some important challenges in terms of fully capturing the real-world complexities and supporting reproducibility in the Computing Continuum. For example, it is increasingly difficult to model the heterogeneity and volatility of Edge devices or to assess the impact of the inherent complexity of hybrid Edge-Cloud deployments on performance. At this stage, experimental evaluation remains the main approach to gain accurate insights of performance metrics, to build precise approximations of the expected behavior of large-scale FL applications on the Computing Continuum, and to eventually validate whether they can be effectively scaled and applied outside of a controlled setting. The gap between simulations and real-world deployments in FL is discussed in [13] and showed to be critical. For instance, the FL algorithms proposed in [42] targeting the communication latency and validated on a Raspberry Pi cluster (16 nodes) already highlight the difference between simulation and a small cluster to be significant.

2.4 Problem Statement: Dynamic Adaptation for Federated Learning

The term "*dynamic adaptation*" is broadly used by the ML and DL communities to designate in-model adjustments for improved learning outcomes based on performance feedback [29]. In the context of our work, focusing on addressing system variance, we define dynamic adaptation as the ability of FL systems to *modify operational strategies in real-time in response to environmental changes*.

To the best of our knowledge, existing FL strategies are agnostic to dynamic changes, as per this definition of dynamic adaptation, and they fail to guarantee accuracy or significantly degrade efficiency. The key reason for this is that some optimal algorithms, or parameter and hyperparameter assignments at the current instant can be suboptimal in the future.

3 MOTIVATION: WHY DYNAMICALLY ADAPT

In this section we briefly recall the main factors of variation for FL deployments across the Computing Continuum, and we illustrate with some real-life use-cases how detecting and then reacting to this variability could improve the overall FL process.

3.1 Dynamics of Federated Learning Environments

Several factors are likely to evolve during the FL execution across Edge-Cloud deployments. We discuss below the most important ones.

Variable network conditions. FL's reliance on ongoing communication means it is vulnerable to network issues [32]. For example, heavy traffic can slow down the exchange of model updates, affecting the training process. In the context of 5G networks, the speed experienced by a device can fluctuate significantly depending on its location relative to the nearest 5G tower. When a device is in close proximity to a tower and has an unobstructed path to it, speeds can reach up to 1000 MB/s. However, these speeds tend to decrease as the distance to the tower increases or if physical barriers disrupt the direct line of sight, impacting the signal strength and quality.

Changing data distributions. The data each client contributes to FL can change over time, either because they collect new types of data or their data collection methods evolve. This continuous change demands that the FL process adjusts to keep the global model accurate and relevant [10]. Several solutions to tackle this problem were proposed in the context of Continual Learning [8], essentially by training the DL model incrementally (i.e., the training proceeds with relatively inexpensive updates to the model's parameters based on just the new data samples). Unfortunately, this approach can cause the accuracy of the DL model to deteriorate quickly—a phenomenon known as *catastrophic forgetting* [24]. Specifically, the training introduces a bias in favor of new samples, effectively causing the model to reinforce recent patterns at the expense of previously acquired knowledge. This is an active field of research in the domain of DL, however, its impact on FL has not yet been studied.

Fluctuating client participation. The number of clients taking part in FL can change dramatically over time, influenced by factors

such as device availability, user choices, and technical limitations. Remember that clients are typically deployed at the Edge, subject to high churn and volatility. The study by Bonawitz et al. [3] illustrates the fluctuating availability of devices for FL over a 24-hour period, highlighting a significant variance from a peak of 6000 devices at midnight to approximately 1000 devices at noon. This could potentially impact the accuracy of the FL, by skewing the federated model towards the data characteristics present during high participation periods, potentially at midnight, and underrepresenting the data characteristics of lower participation periods, like noon. The study also notes that client dropout rates are higher during daytime compared to nighttime, this trend is attributed to changes in device eligibility as users are more likely to interact with their devices during the day. Not only does the number of available clients vary, but their characteristics change as well, highlighting the importance of considering time-dependent variations in FL strategies.

3.2 Performance Optimization with Adaptive Strategies

Detecting and mitigating the impact of such variability could potentially improve the performance of FL systems in several dimensions:

- **Accuracy enhancement:** FL strategies can improve the representativeness and accuracy of the global model by adjusting to the dynamic nature and diversity of data [28].
- **Efficiency improvement:** Optimizing how and when clients communicate based on current network conditions and available resources can reduce training time and resource usage [40].
- **Robustness increase:** Implementing adaptive mechanisms can make FL systems resilient to potential issues such as outliers, security threats, and system failures, maintaining the integrity of the learning process [26, 39].

Several real-life use-cases highlight the importance of adaptability in FL.

Healthcare monitoring. In a network of healthcare facilities sharing data to predict patient outcomes, the variability in incoming patient data can vary widely with seasonal diseases or outbreaks. FL systems must adapt to these changes by adjusting data sampling rates, or model aggregation strategies to ensure timely model convergence without compromising patient privacy or hospital operations [38].

Traffic management systems. Traffic management systems leverage FL to improve traffic flow and safety by learning from data collected across sensors and cameras. Variables such as traffic conditions, sensor functionality, and the quality of data are prone to sudden changes due to unforeseen events like accidents, adverse weather, or infrastructure malfunctions. Adapting client selection and communication strategies can help maintain system performance even under these conditions [9].

Autonomous vehicles. For autonomous vehicles, FL is employed to continually improve driving algorithms based on data from vehicles on the road. These vehicles face rapidly changing conditions like weather, traffic and road conditions. Additionally, the computational capacity for training can fluctuate, as vehicles must give

precedence to navigation tasks and safety. FL needs to adapt dynamically, deciding which data to focus on, synchronization timing and frequency, to improve learning efficiently without affecting the vehicle primary tasks [27].

Although these examples are not exhaustive, the FL patterns they leverage can be generalized in other contexts (i.e., speech and vision, climate science, cancer research, to name a few). Overall, the above examples motivate the need for FL approaches that can model future dynamic adaptation and account for data and system uncertainty while optimizing for both accuracy and efficiency.

4 SUPPORTING USER-DEFINED REACTION: WHAT, WHEN AND HOW TO ADAPT

While dynamic adaptation is a key enabler for efficient FL training, improper changes to the system environment or the FL settings can adversely impact convergence properties and degrade model accuracy. Thus, we argue that FL strategies should support user-defined dynamic adaptation to avoid affecting training accuracy. This means tuning *several* settings and parameters mainly because no single adaptive technique (i.e., single parameter) works consistently well across all datasets and throughout the entire Computing Continuum.

4.1 Adaptation Dimensions

In this context, adjustments can be made across several dimensions:

- **Aggregation Algorithm:** This involves the techniques used to merge client updates into the global model. Adapting this algorithm is essential for dealing with data diversity and varying client reliability. Techniques like weighted averaging [36] prioritize updates from clients with extensive or higher-quality data, while robust aggregation [26] methods aim to reduce the influence of outliers or malicious inputs.
- **Client Selection Method:** Adjusting how clients are chosen for training rounds can optimize the use of resources and improve the efficiency of the learning process. This selection can be adapted based on factors such as network conditions, the quality or amount of data from clients, their past impact on model accuracy, or their computational capabilities [12].
- **Communication Frequency:** The rate at which updates are exchanged between clients and the server affects the speed of model convergence and the overall communication burden. Modifying this frequency in response to current network conditions or the pace of model improvement can make training more efficient [25].

4.2 Decision Criteria

The decision on **when** to react to changes is guided by various conditions:

- **Network and Resource Conditions:** Fluctuations in network speed and resource availability can dictate adjustments in how often communications happen and which clients are included, aiming for efficient bandwidth and

resource usage and minimizing learning interruptions. For example, network congestion caused by a traffic jam might necessitate reducing the frequency of communications between the server and clients to alleviate bandwidth strain.

- **Data Heterogeneity:** The variation in data among clients may require changes in the aggregation algorithm to ensure the global model is effective and representative for all participants. As Li et al. [20] showed, there is not a single FL algorithm that consistently outperforms the other algorithms or has a good performance in all settings, hence finding ways to dynamically match clients evolving data distribution can significantly enhance system performance.
- **Model Performance Metrics:** Monitoring performance indicators like accuracy, loss, or other validation measures helps pinpoint when and what kind of adaptations are necessary to enhance learning outcomes. Detecting a notable decline in accuracy, for example, may prompt the implementation of more robust methods designed to counter potential security threats.

4.3 Algorithmic Approaches

The strategies enabling dynamic FL adaptation fall broadly into two categories.

Reinforcement Learning: RL strategies can refine adaptation measures based on performance feedback [35]. For instance, an RL agent could determine optimal times to adjust communication frequency, striking a balance between network usage and learning efficiency.

Heuristic-Based Approaches: These strategies use straightforward rules or heuristics—derived from empirical data or theoretical knowledge—to inform adaptation decisions. For example, the system could automatically extend communication intervals if network latency reaches a predefined threshold; another example could be a heuristic algorithm to approximate the optimal client selection as proposed in [41].

5 EVALUATION

For this preliminary evaluation, out of the 3 main factors (identified in Section 3.1) affecting the FL across the continuum, we focus on the variable network conditions and the fluctuating client participation. Our objective is to highlight their crucial impact on the FL operational efficiency and the importance of dynamic adaptation to mitigate such effects. Hence, the evaluation seeks to answer the following questions:

- How are the state-of-the-art FL approaches (i.e., FedAvg) impacted by network and client variance in the absence of dynamic adaptation?
- What is the relationship (if any) between the system variability (i.e., number of participating clients, network conditions) and the training time (i.e., the duration required to complete a single round of FL)?

5.1 Experimental Setup

We use a total of 25 nodes from 2 distant sites of Grid’5000 [5], a large-scale testbed for experiment-driven research: 5 nodes from the Lille site are equipped with 2 AMD EPYC 7301 (16 cores/CPU)

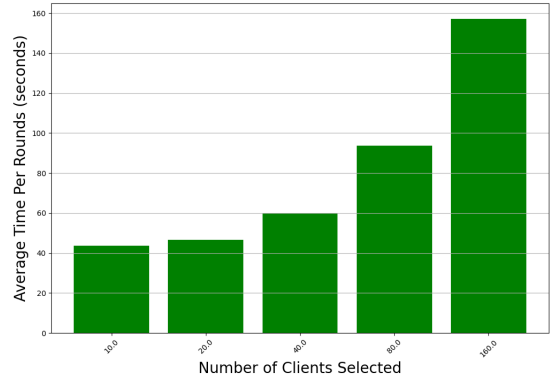


Figure 2: Time per FL round with varying number of participating clients.

each, and 20 nodes on the Rennes site are equipped with 2 Intel Xeon E5-2630 v3 (8 cores/CPU) each. We deploy 500 clients for the federated process (20 clients per node). Each node is given access to an Ethernet configuration with a 10GBps rate.

We leverage the E2Clab framework [30] to automatize our experiments. E2Clab provides tools and functionalities to streamline the deployment and the execution of scientific workflows on Edge-Cloud computing environments. Its integration into our experimental setup allows for orchestration of training processes across distributed nodes.

5.2 Methodology

Evaluation Scenario. In our study, we focus on FedAvg [25], the standard approach for FL with weighted averaging of the client updates to build the global model update. We use FedAvg to train a ResNet-18 network [15] on the CIFAR10 dataset [19]. We rely on the Flower framework [2] for easy setup and deployment of the FL system across multiple nodes.

In each FL round, 10 clients are randomly selected from the pool of 500 clients to participate in the training process. This design mirrors common operational scenarios in FL, where only a subset of available devices participates in each training iteration, balancing computational efficiency and model performance. To evaluate the system performance under different network conditions, we use the *netem* [16] tool to emulate specific network latencies and bandwidth scenarios across the distributed nodes.

Performance Metrics. The primary metric for our investigation is the total duration of a single FL training round, comprising all phases of the FL process: (1) the distribution of the global model to the selected clients, (2) local training on client devices, (3) the transmission of model updates back to the server, and (4) the aggregation of these updates using the FedAvg algorithm. Measuring the total round duration provides a view of the system performance, highlighting the influence of network conditions on the efficiency of FL training.

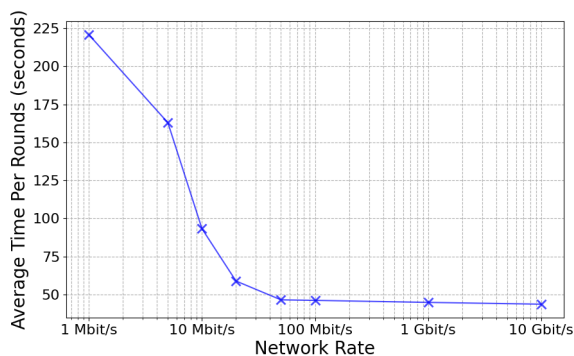


Figure 3: Time per FL round with varying bandwidth.

5.3 Impact of Client Volatility

We vary the number of clients participating in the federation from 10 to 160 and report the time per round in Figure 2. The results show a progressive increase in the time per round as more clients are included in the federation. This trend suggests a potential bottleneck, particularly when multiple clients reside on the same node, slowing down the training process.

In this context, a notable finding is the minimal difference in time per round between selecting 10 and 20 clients. This observation highlights the potential for scalability of FL systems under optimal conditions. Specifically, if the computational resources and network bandwidth are adequate, increasing the number of participating clients does not significantly impact on the training rounds.

5.4 Impact of Network Quality

In the next series of experiments we assess the impact of the network degradation (with different available bandwidth and latency) on the FL training time.

Varying available bandwidth. In the experiment depicted in Figure 3, the network rate ranges from high-speed connections of 10 Gbit/s down to 1 Mbit/s. At low network speeds of 100 Kbit/s, the system ceased to function effectively with the server struggling to maintain or even to establish stable connections with clients, highlighting the role of network quality in maintaining FL operations. At the lower end of the network rate (1 Mbit/s), we observe a significant increase in the time per round. Conversely, at higher network rates, particularly as we approach Gigabit speeds, the time per round reaches a plateau. This suggests that beyond a certain point, improvements in the network speed yield diminishing returns on training speed, likely due to other limiting factors, such as computational latency at the client or server level.

Varying network latency. In our FL context, we define latency as the one-way, end-to-end delay occurring between the server and the client. Figure 4 shows the time per round when the latency changes from 2ms to 100ms. Similar to what we experienced with low network speeds, the system ceased to function effectively at higher latency, the server encountering significant difficulties in establishing and sustaining connections with each client. We observe that even small increments in latency can result in non-negligible

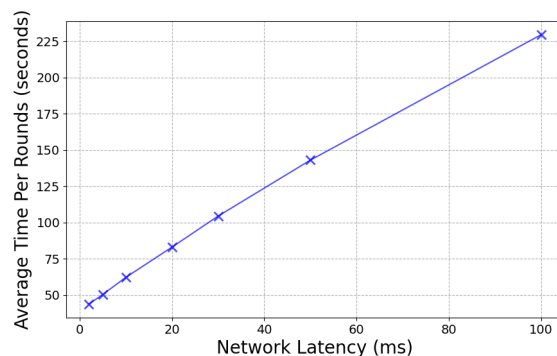


Figure 4: Time per FL round with varying network latency.

delays in round completion. This sensitivity to latency variations is an important consideration for the deployment of FL systems, especially in environments where network conditions are variable or cannot be strictly controlled.

These preliminary results demonstrate the sensitivity of existing FL systems to variations in network quality, including bandwidth and latency. As state-of-the-art FL approaches like FedAvg lack support for dynamic adaption, there is an urgent need to fill this gap by designing FL systems able to: (1) detect in real-time changes in the deployment environment (e.g., network conditions), and (2) react swiftly by updating system parameters (e.g., frequency or number of clients selected each round) or changing the underlying algorithms.

6 DISCUSSION

Several take-aways emerge for the FL community regarding the early design and development of strategies for dynamically adapting FL systems.

6.1 Emphasize System Adaptability

The FL community should prioritize building systems with inherent flexibility. This means designing FL frameworks capable of smooth and seamless modification of important operational aspects, such as the client selection method or the communication protocol, based on real-time assessments of the environment, like the network conditions or the computational constraints.

6.2 Monitor and Respond in Real-Time

Adapting to the system environment variability is crucial. Continuous monitoring of network conditions and system performance should form the backbone of dynamic adaptation strategies. For example, by incorporating network sensing technologies and predictive analytics into FL systems, it could be possible to preemptively adjust operations *before* network degradation significantly impacts performance. To ensure the dynamic adaptation strategies remain efficient without burdening the FL process, these monitoring mechanisms should be non-intrusive. For example, leveraging lightweight communication protocols like UDP for monitoring tasks can minimize the overhead. This approach ensures that the essential FL

operations remain unaffected while still gaining the benefits of responsive adaptation based on environmental feedback.

6.3 Engage in Cross-Disciplinary Research

Addressing the challenges of dynamic adaptation in FL will benefit from a cross-disciplinary approach, drawing insights from fields such as network engineering, machine learning, distributed computing and even behavioral sciences to understand and predict system and participant behavior.

7 CONCLUSION

Federated Learning has advanced to the point where its application is pervasive, spanning diverse domains. However, with its increased adoption across the Computing Continuum comes the need to effectively mitigate the issues related to the inherent dynamicity of such infrastructures. In this paper, we showed how existing FL systems degrade efficiency when faced with variable execution conditions. To address these challenges, we proposed to add dynamic adaptation support in FL and provided some initial guidelines into this direction.

Looking forward, we plan to extend our experimental evaluation with a wider range of real-world scenarios and testing different FL methods. We also plan to study how different existing tools for monitoring and responding to environment conditions could be integrated in a lightweight fashion into existing FL strategies.

ACKNOWLEDGMENTS

This work was supported by a French government grant, managed by the *Agence Nationale de la Recherche* (ANR) under the France 2030 program, reference ANR-23-PECL-0007 (PEPR CLOUD - STEEL project). Experiments presented in this paper were carried out using the French Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see <https://www.grid5000.fr>).

REFERENCES

- [1] Guanming Bao and Ping Guo. 2022. Federated learning in cloud-edge collaborative architecture: key technologies, applications and challenges. *Journal of Cloud Computing* 11, 1 (Dec. 2022), 94. <https://doi.org/10.1186/s13677-022-00377-4>
- [2] Daniel J. Beutel, Taner Topal, Akhil Mathur, Xinchu Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Kwing Hei Li, Titouan Parcollet, Pedro Porto Buarque de Gusmão, and Nicholas D. Lane. 2022. Flower: A Friendly Federated Learning Research Framework. <https://doi.org/10.48550/arXiv.2007.14390> [cs, stat].
- [3] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečný, Stefano Mazzocchi, H. Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. 2019. Towards Federated Learning at Scale: System Design. <https://doi.org/10.48550/arXiv.1902.01046> [cs, stat].
- [4] Sofiane Bouaziz, Hadjer Benmeziane, Youcef Imine, Leila Hamdad, Smail Niar, and Hamza Ouarnoughi. 2023. FLASH-RL: Federated Learning Addressing System and Static Heterogeneity using Reinforcement Learning. In *2023 IEEE 41st International Conference on Computer Design (ICCD)*. 444–447. <https://doi.org/10.1109/ICCD58817.2023.00074> ISSN: 2576-6996.
- [5] F. Cappello, E. Caron, M. Dayde, F. Desprez, Y. Jegou, P. Prinet, E. Jeannot, S. Lanteri, J. Leduc, N. Melab, G. Mornet, R. Namyst, B. Quetier, and O. Richard. 2005. Grid'5000: a large scale and highly reconfigurable grid experimental testbed. In *The 6th IEEE/ACM International Workshop on Grid Computing*, 2005. 8 pp.–. <https://doi.org/10.1109/GRID.2005.1542730> ISSN: 2152-1093.
- [6] Shen Chen, Zekai Lin, and Jing Ma. 2023. The Effect of Hyper-parameters in Model-contrastive Federated Learning Algorithm. In *2023 IEEE International Conference on Sensors, Electronics and Computer Engineering (ICSECE)*. 1170–1174. <https://doi.org/10.1109/ICSECE58870.2023.10263504>
- [7] Zihan Chen, Jundong Li, and Cong Shen. 2024. Personalized Federated Learning with Attention-Based Client Selection. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 6930–6934. <https://doi.org/10.1109/ICASSP48485.2024.10447362> ISSN: 2379-190X.
- [8] Marcos F. Criado, Fernando E. Casado, Roberto Iglesias, Carlos V. Regueiro, and Senén Barro. 2022. Non-IID data and Continual Learning processes in Federated Learning: A long road ahead. *Information Fusion* 88 (Dec. 2022), 263–280. <https://doi.org/10.1016/j.inffus.2022.07.024>
- [9] Ahmet M. Elbir, Burak Soner, Sinem Çöleri, Deniz Gündüz, and Mehdi Bennis. 2022. Federated Learning in Vehicular Networks. In *2022 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*. 72–77. <https://doi.org/10.1109/MeditCom55741.2022.9928621>
- [10] Amir Hossein Estiri and Muthucumar Maheswaran. 2021. Attentive Federated Learning for Concept Drift in Distributed 5G Edge Networks. <https://doi.org/10.48550/arXiv.2111.07457> arXiv:2111.07457 [cs].
- [11] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. Personalized Federated Learning: A Meta-Learning Approach. <https://doi.org/10.48550/arXiv.2002.07948> [cs, math, stat].
- [12] Yann Fraboni, Richard Vidal, Laetitia Kameni, and Marco Lorenzi. 2022. A General Theory for Client Sampling in Federated Learning. <http://arxiv.org/abs/2107.12211> arXiv:2107.12211 [cs].
- [13] Hend K. Gedawy, Khaled A. Harras, Thang Bui, and Temoor Tanveer. 2023. RealFL: A Realistic Platform for Federated Learning. In *Proceedings of the Int'l ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM '23)*. Association for Computing Machinery, New York, NY, USA, 313–317. <https://doi.org/10.1145/3616388.3623799>
- [14] Filip Hanzely and Peter Richtárik. 2021. Federated Learning of a Mixture of Global and Local Models. <https://doi.org/10.48550/arXiv.2002.05516> arXiv:2002.05516 [cs, math, stat].
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. <https://doi.org/10.48550/arXiv.1512.03385> arXiv:1512.03385 [cs].
- [16] Stephen Hemminger. 2005. Network emulation with NetEm. *Linux Conf Au* (May 2005).
- [17] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. 2021. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. <https://doi.org/10.48550/arXiv.1910.06378> [cs, math, stat].
- [18] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2017. Federated Learning: Righties for Improving Communication Efficiency. <https://doi.org/10.48550/arXiv.1610.05492> [cs].
- [19] Alex Krizhevsky. 2012. Learning Multiple Layers of Features from Tiny Images. *University of Toronto* (May 2012).
- [20] Qinqin Li, Yiqun Diao, Quan Chen, and Bingsheng He. 2022. Federated Learning on Non-IID Data Silos: An Experimental Study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. 965–978. <https://doi.org/10.1109/ICDE53745.2022.00077> ISSN: 2375-026X.
- [21] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine* 37, 3 (May 2020), 50–60. <https://doi.org/10.1109/MSP.2020.2975749> Conference Name: IEEE Signal Processing Magazine.
- [22] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2020. On the Convergence of FedAvg on Non-IID Data. <https://doi.org/10.48550/arXiv.1907.02189> [cs, math, stat].
- [23] Bingyan Liu, Nuoyan Lv, Yuanchun Guo, and Yawen Li. 2023. Recent Advances on Federated Learning: A Systematic Survey. <https://doi.org/10.48550/arXiv.2301.01299> arXiv:2301.01299 [cs].
- [24] Michael McCloskey and Neal J. Cohen. 1989. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In *Psychology of Learning and Motivation*, Gordon H. Bower (Ed.). Vol. 24. Academic Press, 109–165. [https://doi.org/10.1016/S0079-7421\(08\)60536-8](https://doi.org/10.1016/S0079-7421(08)60536-8)
- [25] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2016. Communication-Efficient Learning of Deep Networks from Decentralized Data. <https://doi.org/10.48550/arXiv.1602.05629>
- [26] El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. 2018. The Hidden Vulnerability of Distributed Learning in Byzantium. <https://doi.org/10.48550/arXiv.1802.07927> arXiv:1802.07927 [cs, stat].
- [27] Anh Nguyen, Tuong Do, Minh Tran, Binh X. Nguyen, Chien Duong, Tu Phan, Erman Tjiputra, and Quang D. Tran. 2022. Deep Federated Learning for Autonomous Driving. In *2022 IEEE Intelligent Vehicles Symposium (IV)*. 1824–1830. <https://doi.org/10.1109/IV51971.2022.9827020>
- [28] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H. Brendan McMahan. 2021. Adaptive Federated Optimization. <https://doi.org/10.48550/arXiv.2003.00295>

- [29] Elsa Rizk, Stefan Vlaski, and Ali H. Sayed. 2020. Dynamic Federated Learning. <https://doi.org/10.48550/arXiv.2002.08782> arXiv:2002.08782 [cs, stat].
- [30] Daniel Rosendo, Pedro Silva, Matthieu Simonin, Alexandru Costan, and Gabriel Antoniu. 2020. E2Clab: Exploring the Computing Continuum through Repeatable, Replicable and Reproducible Edge-to-Cloud Experiments. In *Cluster 2020 - IEEE International Conference on Cluster Computing*, Kobe, Japan, 1–11. <https://doi.org/10.1109/CLUSTER49012.2020.00028>
- [31] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. 2019. Clustered Federated Learning: Model-Agnostic Distributed Multi-Task Optimization under Privacy Constraints. <https://doi.org/10.48550/arXiv.1910.01991> arXiv:1910.01991 [cs, stat].
- [32] Giorgio Severi, Matthew Jagielski, Gökberk Yar, Yuxuan Wang, Alina Oprea, and Cristina Nita-Rotaru. 2022. Network-Level Adversaries in Federated Learning. In *2022 IEEE Conference on Communications and Network Security (CNS)*, 19–27. <https://doi.org/10.1109/CNS56114.2022.9947237>
- [33] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. 2017. Federated Multi-Task Learning. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc. <https://proceedings.neurips.cc/paper/2017/hash/6211080fa89981f66b1a0c9d55c61d0f-Abstract.html>
- [34] Alysia Ziyang Tan, Han Yu, Lizhen Cui, and Qiang Yang. 2022. Towards Personalized Federated Learning. *IEEE Transactions on Neural Networks and Learning Systems* (2022), 1–17. <https://doi.org/10.1109/TNNLS.2022.3160699>
- [35] Hao Wang, Zakhary Kaplan, Di Niu, and Baochun Li. 2020. Optimizing Federated Learning on Non-IID Data with Reinforcement Learning. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 1698–1707. <https://doi.org/10.1109/INFOCOM41043.2020.9155494> ISSN: 2641-9874.
- [36] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. 2020. Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization. <https://doi.org/10.48550/arXiv.2007.07481> arXiv:2007.07481 [cs, stat].
- [37] Changxin Xu, Yuxin Qiao, Zhanxin Zhou, Fanghao Ni, and Jize Xiong. 2024. Enhancing Convergence in Federated Learning: A Contribution-Aware Asynchronous Approach. <https://doi.org/10.48550/arXiv.2402.10991> arXiv:2402.10991 [cs].
- [38] Jie Xu, Benjamin S. Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. 2021. Federated Learning for Healthcare Informatics. *Journal of Healthcare Informatics Research* 5, 1 (March 2021), 1–19. <https://doi.org/10.1007/s41666-020-00082-4>
- [39] Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter Bartlett. 2021. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. <https://doi.org/10.48550/arXiv.1803.01498> arXiv:1803.01498 [cs, stat].
- [40] Zichao Zhao, Junjuan Xia, Lisheng Fan, Xianfu Lei, George K. Karagiannidis, and Arumugam Nallanathan. 2022. System Optimization of Federated Learning Networks With a Constrained Latency. *IEEE Transactions on Vehicular Technology* 71, 1 (Jan. 2022), 1095–1100. <https://doi.org/10.1109/TVT.2021.3128559> Conference Name: IEEE Transactions on Vehicular Technology.
- [41] Jingjing Zheng, Kai Li, Eduardo Tovar, and Mohsen Guizani. 2021. Federated Learning for Energy-balanced Client Selection in Mobile Edge Computing. In *2021 International Wireless Communications and Mobile Computing (IWCMC)*, 1942–1947. <https://doi.org/10.1109/IWCMC51323.2021.9498853> ISSN: 2376-6506.
- [42] Ligeng Zhu, Hongzhou Lin, Yao Lu, Yujun Lin, and Song Han. 2021. Delayed Gradient Averaging: Tolerate the Communication Latency for Federated Learning. In *Advances in Neural Information Processing Systems*, Vol. 34. Curran Associates, Inc., 29995–30007. <https://proceedings.neurips.cc/paper/2021/hash/fe03d48253286a798f5116ec00e99b2b-Abstract.html>