Inria / hive Challenge

# CUPSELI: Collaborative Unified Platform for a Scalable and Efficient Learning Infrastructure


ARGO and MIMOVE, Inria Paris
Kevin Scaman <kevin.scaman@inria.fr>
Nikolaos Georgantas <Nikolaos.Georgantas@inria.fr>

COAST, Inria Nancy – Grand Est
Claudia-Lavinia Ignat <claudia.ignat@inria.fr>
Thomas Lambert <thomas.lambert@inria.fr>

MAGELLAN, STACK and WIDE, Inria Centre at Rennes University
David Bromberg <David.Bromberg@inria.fr>
Philippe Chartier <philippe.chartier@inria.fr>
Davide Frey <davide.frey@inria.fr>
Shadi Ibrahim <shadi.ibrahim@inria.fr>
Mohammed Lemou <mohammed.lemou@univ-rennes1.fr>
Guillaume Rosinosky <guillaume.rosinosky@inria.fr>
Mario Sudhölt <mario.sudholt@inria.fr>

OCKHAM, Inria Centre of Lyon
Remi Gribonval <remi.gribonval@inria.fr>
Elisa Riccietti <elisa.riccietti@ens-lyon.fr>

COATI and NEO, Inria Centre at Université Côte d'Azur
Frédéric Giroire <frederic.giroire@cnrs.fr>
Giovanni Neglia <giovanni.neglia@inria.fr>
Samir Perlaza <samir.perlaza@inria.fr>
Chuan Xu <chuan.xu@inria.fr>

TADAAM and TOPAL, Inria Centre at the University of Bordeaux
Olivier Beaumont <olivier.beaumont@inria.fr>
Alexandre Denis <Alexandre.Denis@inria.fr>
Lionel Eyraud-Dubois <Lionel.Eyraud-Dubois@inria.fr>
Julia Gusak <Yulia.Gusak@inria.fr>
Thomas Herault <herault@icl.utk.edu>
Laércio Lima Pilla <laercio.pilla@inria.fr>
Philippe Swartvagher <philippe.swartvagher@inria.fr>

hive
Antoine Clerget <antoine.clerget@hivenet.com>
Alexadru Dobrila <alexandru.dobrila@hivenet.com>

# 1 INTRODUCTION

## 1.1 Context

hive offers a highly original data storage architecture in which data is stored in a distributed and secure manner on the spare storage resources of participants, based on a peer-to-peer structure. This structure naturally ensures scalability, resilience and voluntary sharing of data between users. Another advantage of this architecture is its positive impact on carbon emissions by using and enhancing existing resources rather than building new datacenters, which have a very high impact.

A first challenge, Alvearium [1], between Inria and hive, explored the possibility of considering mutable data and enabling its collaborative editing, while ensuring security, consistency and privacy, and guaranteeing low response times.

The aim of this new challenge between hive and Inria is to extend storage issues to those of computation. This objective is particularly relevant in the present context. Indeed, on the one hand, from a hardware point of view, while GPUs are very hard to obtain, there are significant resources available at users' premises that are for the most part idle, potentially providing very significant computational power. On the other hand, we are witnessing the emergence of new applications that naturally fit well with distributed execution, such as the inference of large-scale artificial intelligence models that require a large number of independent tasks to be executed. These applications are well suited to distributed execution on interconnected computing resources, even if the network is significantly less powerful than that of a supercomputer. Although the execution of highly coupled numerical simulation codes or kernels is probably beyond the capabilities of such a platform, the execution of fine-tuning tasks and even the training of large models are within the scope of this challenge.

## 1.2 Challenges

In this context, the main issues and difficulties to be considered in the challenge can be clearly identified and are in particular related to the specific characteristics of the hive Compute platform.

— First of all, the computational resources are typically one or two generations older than the latest GPUs on which large language models are typically trained. It is therefore crucial to implement algorithmic strategies to compensate for the low memory capacity of the GPUs involved, in particular by using more resources, by performing more computations and/or by using storage resources other than GPU memory.

— The second difficulty is related to the interconnection of computational resources, which is slow compared to what can be found, for example, in a supercomputer. While this limitation is acceptable in the context of inference or processing large batches of independent tasks, such as in the banking sector, it can become prohibitive in the context of training large models. Again, the solution lies in a fundamental change in the optimisation algorithms, moving towards more distributed versions that are more efficient in terms of data exchange, even if this means being slightly more expensive in terms of computation.

— The final constraint refers to data security. While hive's storage infrastructure guarantees the privacy of data (typically training data), it is crucial that this privacy is also guaranteed when performing computations on this data. Depending on the level of security to be guaranteed (which depends on the application context), algorithmic solutions ranging from federated learning

1. https://project.inria.fr/alvearium/

to homomorphic encoding, system solutions based on containment, and hardware solutions based on confidential computing can be considered.

## 1.3 Target Computing Platforms

To address these issues, we will consider computing platforms of increasing complexity, with different levels of distribution, sharing, and scale, corresponding to the different technological solutions proposed by hive. Depending on the type of platform considered, the nature of the applications (requiring more or less data exchange, for instance) and the level of privacy required will need to be adapted, but the goal is to propose a set of algorithmic and system solutions based on available hardware that allow a wide range of applications to be implemented, possibly at the expense of additional computation.

— The first scale corresponds to private corporate or administrative networks (local private hive Nets), which can be single- or multi-site. In this context, data security and privacy are less critical since the resources are corporate, but issues related to resource type (heterogeneity of resources, low storage capacity) and connectivity must be taken into account.

— The second scale corresponds to using the peer-to-peer storage and computing structure proposed by hive. Compared to the scale of private enterprise networks, strategies are necessary to ensure data privacy and to manage the intermittent availability of resources (elasticity, fault tolerance, malleability). For efficiency and sovereignty reasons, the issue of computing resources allocation is also becoming critical.

— The third, more forward-looking scale corresponds to the use of highly heterogeneous resources (CPU, GPU, TPU, NPU), typically relying on the processors available in tablets and phones, in addition to computers. This scale exacerbates the challenges identified at previous scales.

## 1.4 Goals

The main objective of Cupseli is to demonstrate the feasibility of porting applications that are as complex as possible (in terms of more communications, more interdependencies between tasks, more computations) onto platforms that are as complex as possible (in terms of size, heterogeneity and volatility of resources), while ensuring the highest possible quality of service (in terms of privacy and response times).

More specifically, our aim is to demonstrate the ability to perform large model fine tuning on several dozen heterogeneous, distributed and volatile resources, with an overall performance of around 50% of the aggregated power of the resources, without degrading accuracy.

## 1.5 Complementarity between hive and Inria

hive brings a wealth of industrial experience, both in terms of system and application expertise and in the management of a large distributed storage infrastructure. In addition, hive provides the opportunity to experiment with distributed technologies at very large scale, enabling the validation of algorithmic solutions and software prototypes developed by Inria teams.

Within the framework of this Challenge, Inria is bringing together highly complementary teams with rich and diverse expertise in cloud computing, distributed algorithms and training of large AI models. More specifically, Inria brings the expertize of its teams in (i) efficient training, fine-tuning and inference (**Coati, Mimove, Topal, Argo** teams), compression (**Coati, Ockham, Topal** teams),

federated learning (**Coati, Neo, Topal** teams), Distributed training (**Argo, Neo**), in (ii) Optimized allocation of computations and services (**Coast, Magellan, Topal** teams), optimized communications (**Tadaam, Topal** teams) and large scale distributed systems (**Mimove, Wide** teams) and (iii) in the management of the dynamic nature of applications (replication, migration, elasticity, malleability, reconfiguration) (**Coast, Magellan, Topal** teams) and Containerization, Confidential Computing (**Wide, Stack** teams).

# 2 GENERAL ORGANIZATION OF THE CHALLENGE

## 2.1 Scientific Organization

For the sake of readability, we propose to organize the challenge around three axes that capture the identified challenges specific to the hive platform:

— Axis 1, coordinated by Julia Gusak (Section 2.1.1): **Frugality**, in terms of memory and data stored and exchanged more than in terms of computation, due to the characteristics of the participating resources and their interconnection,

— Axis 2, coordinated by Davide Frey (Section 2.1.2): **Security and Privacy**, necessary for some applications in the context of a distributed execution among volunteer participants,

— Axis 3, coordinated by Giovanni Neglia (Section 2.1.3): **Volatility and Fault Tolerance**, in an environment where resources can leave or join the platform at any time.

This organization into axes will allow for more effective and relevant scientific animation. Nevertheless, the project must be seen as a global effort, and all the addressed issues contribute to the shared objective of enabling the efficient execution of computational workloads, essentially training and inference, on the hive platform. The fact that the set of targeted applications is very small is an advantage for strengthening the collaboration between the axes. In Section 3, we highlight the strong links that exist between the proposed topics, which go well beyond the axes.

In the following summary tables (one for each axis), we propose a set of keywords and indicate which keywords are associated with each proposed PhD Thesis, postdoctoral and engineering position.

| Axis 1 | PhD 3.1.1 | PhD 3.1.2 | PostDoc 3.1.3 | Eng 3.1.4 |
|---|---|---|---|---|
| Memory Frugality | ✓ | | ✓ | ✓ |
| Data and Comm. Frugality | ✓ | ✓ | ✓ | ✓ |
| Security | | | | |
| privacy | | | | |
| Volatility | ✓ | ✓ | | |
| Heterogeneity | ✓ | ✓ | | ✓ |
| Training | ✓ | ✓ | ✓ | ✓ |
| Inference | ✓ | ✓ | ✓ | |
| Fault Tolerance | ✓ | ✓ | | |

| Axis 2 | PhD 3.2.1 | PhD 3.2.2 | PhD 3.2.3 | Eng 3.2.4 | PhD 3.2.5 |
|---|---|---|---|---|---|
| Memory Frugality | | | | | |
| Data and Comm. Frugality | | | | | |
| Security | ✓ | ✓ | ✓ | ✓ | ✓ |
| privacy | ✓ | ✓ | ✓ | ✓ | ✓ |
| Volatility | | | | | |
| Heterogeneity | | ✓ | ✓ | | ✓ |
| Training | ✓ | ✓ | ✓ | ✓ | ✓ |
| Inference | ✓ | ✓ | ✓ | ✓ | |
| Fault Tolerance | | | | | ✓ |

| Axis 3 | PhD 3.3.1 | PostDoc 3.3.2 | PhD 3.3.3 | PhD 3.3.4 | Eng 3.3.5 |
|---|---|---|---|---|---|
| Memory Frugality | | | ✓ | | ✓ |
| Data and Comm. Frugality | ✓ | ✓ | | ✓ | |
| Security | | | ✓ | | |
| Privacy | | | | | |
| Volatility | ✓ | ✓ | ✓ | ✓ | ✓ |
| Heterogeneity | ✓ | ✓ | ✓ | ✓ | ✓ |
| Training | | | ✓ | ✓ | ✓ |
| Inference | | | ✓ | | ✓ |
| Fault Tolerance | ✓ | ✓ | ✓ | | ✓ |

### 2.1.1 Axis 1: Frugal Training on Dynamic Resources (lead by Julia Gusak)

The first Axis of the Challenge is related to the adaptation of training and inference jobs to an environment in which resource capacities (particularly in terms of memory) are limited, and in which resources are volatile and the computing platform dynamic.

Two main approaches are envisaged for dealing with memory limitations:

1. The first approach consists of performing exactly the same computations, but either (i) trading memory requirements for additional computations or data exchanges (GPU to CPU), or (ii) aggregating the memory of several GPUs to be able to store the model and associated activations. Both are clearly consistent with the characteristics of the hive computing platform. This is the approach followed in the Thesis 3.1.1 and the Engineering position 3.1.4.
2. The second approach consists in working on the data representation and using compression techniques for weights and activations, while controlling the resulting error. This is the approach taken in Postdoc 3.1.3.

To manage the dynamicity of computing resources, it is important to design communication libraries that are robust to the addition and removal of resources. Such robustness is known to be difficult to implement in the general case, but it may be possible to take advantage of the nature of data exchanges induced by a stochastic process like training, to minimize the cost of robustness. This is the approach proposed in Thesis 3.1.2.

### 2.1.2 Axis 2: privacy and security for training, algorithmic, system and hardware approaches (lead by Davide Frey)

The second Axis of Cupseli naturally focuses on security and privacy issues. Indeed, while the distributed data storage layer in HiveNet, as considered in Alvearium, provides data security and

privacy, it is crucial that the latter is preserved throughout the computation, at least in cases where the application context requires it.

In Cupseli, we will consider three main strategies for ensuring data privacy and computational security, focusing in particular on the analysis of induced overhead:

1. The first strategy is purely software-based, and relies on homomorphic encryption to address the privacy risks associated with decentralized model sharing. In this context, it is well known that the cost of multiplication is very high, requiring complementary algorithmic strategies to minimize the number of operations to be performed using homomorphic encryption. This is the subject of the PhD thesis 3.2.1.

2. The second strategy relies on the hardware capabilities of processors to prevent any participant to become a potential attacker. The idea here is to use the capabilities of modern CPUs to achieve the isolation of both AI models and data within Trusted Execution Environments. The PhD Theses 3.2.2 and 3.2.3 and the Engineering position 3.2.4 explore this approach, analyzing both system and algorithmic issues to manage the heterogeneous capabilities of CPUs in terms of Confidential Computing.

3. Finally, a third approach aims to ensure that a participant cannot perform model poisoning, i.e. maliciously degrade the model or install backdoors that can be later exploited. PhD Thesis 3.2.5 aims to develop a theoretical framework to quantify privacy leakage and to mitigate these attacks.

### 2.1.3 Axis 3: Large-Scale Computing on Intermittent Resources (lead by Giovanni Neglia)

The third Axis of Cupseli focuses on scalability and the management of resource volatility, i.e. the fact that participants can leave or join the system at any time. In this context, it is crucial to guarantee system robustness and application correctness, even in the presence of faults. It is also crucial to ensure that load balancing mechanisms enable the efficient use of a collection of inherently heterogeneous resources (in terms of memory, computing speed, network connectivity, etc.). Unlike the first two axes, which focus on training and inference, this axis considers both Big Data and AI applications:

— In the context of Big Data applications, which are generally run in cloud environments, the HiveNet environment poses a number of challenges related to the volatility and heterogeneity of resources. On the one hand, it is essential to consider dynamic system strategies (at runtime) for placing and scheduling calculations. On the other hand, it is crucial to build algorithmic strategies that take into account stragglers and failures and induce minimal computational overhead. These are the approaches followed in the Thesis 3.3.1 and the postdoc 3.3.2.

— In the context of training, the Thesis 3.3.3 aims to use the specific features of training and inference to achieve efficient allocation of data and computation, and make use of the specific parallel opportunities and data compression strategies, while investigating defenses against byzantine behavior and Sybil attacks. The Thesis 3.3.4 focuses on customer volatility, and in particular on using a predictive model of customer availability to optimize system performance.

— Finally, the engineering position 3.3.5 provides a containerization layer for the management of both Big Data and AI workloads that takes into account the specific characteristics of the hive platform.

## 2.2 Timeline and Resource Allocation

In this section, we specify for each proposed PhD Thesis, postdoctoral and engineering position the main location for the non-permanent recruited. Please note, however, that the vast majority of subjects are co-supervised by two Inria teams. For practical reasons, it is important that each non-permanent recruited has a main location and the possibility of spending a few weeks in the other team and in hive offices in Cannes, whatever their status.

| Position | Main location (co-supervision) | 09/2025 | 09/2026 | 09/2027 | 09/2028 |
|---|---|---|---|---|---|
| PhD 3.1.1 | Bordeaux (Nancy) | | | | |
| PhD 3.1.2 | Bordeaux (Bordeaux) | | | | |
| PostDoc 3.1.3 | Lyon (Sophia) | | | | |
| Eng 3.1.4 | Cannes (Bordeaux) | | | | |
| PhD 3.2.1 | Rennes (Rennes) | | | | |
| PhD 3.2.2 | Rennes | | | | |
| PhD 3.2.3 | Nantes | | | | |
| Eng 3.2.4 | Cannes (Nantes) | | | | |
| PhD 3.2.5 | Sophia (Sophia) | | | | |
| PhD 3.3.1 | Nancy (Rennes) | | | | |
| PostDoc 3.3.2 | Rennes | | | | |
| PhD 3.3.3 | Paris (Rennes) | | | | |
| PhD 3.3.4 | Paris (Sophia) | | | | |
| Eng 3.3.5 | Cannes (Nantes) | | | | |

# 3 LIST OF RESEARCH TOPICS

## 3.1 Axis 1: Frugal Training on Dynamic Resources

**Coordination: Julia Gusak (Topal)**

### 3.1.1 Distributed inference (throughput/latency), fine tuning with memory shortage

**Resources:** One PhD Thesis starting 09/25

**Supervision:** Topal (Olivier Beaumont, Lionel Eyraud-Dubois, Julia Gusak, Laércio Lima Pilla) and Coast (Thomas Lambert) for Inria and Mamoutou Diarra for Hive

**Related Topics**: PhD Thesis 3.3.3, PhD Thesis 3.3.4, Engineer 3.1.4

**Context:** We are interested here in the inference of large models (which typically do not fit into the memory of a single GPU), on a hive Net private or hive Net public computing platform, with a target of a few dozen resources. In the context of inference, several issues need to be addressed:

— Each token generation must a priori pass through several GPUs, each storing different parts of the model. There is a problem of model partitioning [1, 2] and also a problem associated with the construction of inference paths to minimize latency (making groups of nearby resources);

— The amount of inference required will naturally vary over time, and the set of resources made available to the computation will also vary. There is therefore a static planning problem (deciding

which resources are likely to participate and storing the models there) and a dynamic problem (how to allocate new requests);

— Depending on the models used, some of the inference tasks are naturally placed on certain resources (because previous tokens have been generated there, for example). In terms of Fault Tolerance, the computation can easily be restarted (knowing which tokens have been generated), but at a high cost. This raises problems of resource allocation depending on resource availability statistics.

**Goals:** The aim of this PhD thesis is to develop a prototype for performing inference in a dynamic environment (due to resource volatility and varying demand):

— for more or less complex models (fitting in GPU memory or not, re-entering models for token generation or not);

— on more or less complex platforms (in terms of heterogeneity, volatility, failures, size, location, etc.).

At this stage, it is difficult to have precise objectives in terms of deployment on hive Compute, and depending on technical difficulties, we may be able to carry out simulations or emulation. We have not planned to consider issues related to security (apart from fault tolerance).

In the context of this topic, we also consider it relevant to look at data-stream processing (DSP), whose execution scheme is very similar to our problem. More precisely, DSPs are models representing the continuous processing of data (akin to token generation), which passes through various operators (replicated as required) represented by a DAG. There is a large literature on the placement of operators in such models, with optimization of various metrics [3] (latency, throughput, communications, scaling, fault tolerance, etc.), although most of it concerns the case of execution in clouds [4] (fewer resource constraints and less heterogeneity).

In addition, there are several tools that implement this type of model [5] (Apache Flink or Apache Storm, for example), and studying their characteristics could also serve as a basis for setting up a prototype.

### 3.1.2 Communication primitives for training on Volatile Distributed Platforms

**Resources:** One PhD Thesis starting 09/25

**Supervision:** Topal (Philippe Swartvagher, Thomas Herault) and Tadaam (Alexandre Denis) for Inria and Mamoutou Diarra for Hive

**Related Topics**: PhD Thesis 3.1.1, Engineer 3.1.4

**Context:** This PhD thesis focuses on the network communication aspects of the applications to be run. Network communications will differ from what is traditionally done in an HPC environment for several reasons. Firstly, the machines are interconnected via a conventional Ethernet network, which is less efficient than network technologies dedicated to HPC. Secondly, we must keep in mind that computing resources are not permanently available (for example, machines are more available at night) and are more likely to disappear at any moment. Finally, using machines belonging to different geographical sites creates a network with heterogeneous performance: the latency to communicate between two sites is much higher than within a single site [6].

**Goals:** The objective of this PhD thesis is therefore to explore all the issues related to network communications that emerge in such a context. First of all, it will be necessary to define the most suitable API for enabling machines to communicate under such conditions, in particular whether it is still possible to use MPI, whether it needs to be adapted [7, 8] or whether more distributed models should be considered instead, such as `libp2p` [2]. Once this communication model has been defined, given a set of machines and their interconnection topology, we will consider the necessary adaptations to the communication schemes, in particular in the context of training, in order to minimize communication costs: for example, by using routing algorithms and a distribution of computations and data more suited to the network interconnecting the computing resources. It will also be necessary to be able to detect the loss and the possible arrival of machines and to adapt accordingly, for example by ignoring the contributions of lost machines in the case of data parallelism, or by redistributing data and computations. We will also consider network occupation management in the case where both hive's storage and computing services are present simultaneously on the same networks and machines, in order to maintain satisfactory performance for both services and to dynamically adapt quality of service parameters according to network conditions and user requirements.

### 3.1.3 Exploiting symmetries and harnessing sparsification in modern neural networks

**Resources:** One 2-year postdoc starting 09/26

**Supervision:** Ockham (Elisa Riccietti and Rémi Gribonval), Topal (Julia Gusak) and Coati (Frederic Giroire)

**Related Topics**: PhD Thesis 3.1.1

**Context:** The impressive success of machine learning models and deep neural networks in particular ensured great achievements in several domains. However, as these successes are mainly explained by the use of large models and large datasets, the training and deployment of such models leads to enormous energy consumption and carbon emission. This undermines a democratic access to deep learning, hinders its use in contexts in which resources are limited, and has a negative impact on the environment. It is thus of crucial importance to make deep learning more parsimonious, by reducing its data-hungry nature and its dependence on large deep neural networks. A recent line of works, known as the (Strong) Lottery Ticket Hypothesis [9, 10], has shown that any sufficiently large neural network contains a subnetwork, or winning ticket, reaching a performance close to the one of the large network. This gives an insight on the potential of model size reduction. However, finding the winning ticket is an open area of research. Among the most used approaches to achieve the parsimony goal are quantization[11] and sparsification [12].

**Goals:** The main problem of existing techniques is a loss in model performance. This is usually due to the fact that the distribution of weights and activations breaks after these approximations are done. Strategies exist to deal with the weights distribution,while for the activations the problem is still open. Moreover, the existing techniques do not exploit important symmetries often encountered in such problems, such as neuron permutations and weight rescaling invariances [13]. Finally, unstructured sparsity has been documented to lead to unstable optimization problems [14]. Our main objective is to develop sound approaches, supported by theoretical guarantees, capable of exploiting model symmetries while controlling the distributions of the activation functions to keep the performance drop under control. Besides, we target structured parsimony [15, 16] in order to promote stability of the

2. https://libp2p.io/

developed approaches and to allow for efficient implementations compatible with modern computing architectures.

A promising approach that we will investigate is the approximation of the weights matrices of neural networks by Butterfly matrices [17, 18, 19, 20], quantized [21, 22] or not. Butterfly matrices are very expressive products of sparse matrices and can approximate a large class of dense matrices. While their ability to approximate the linear layers of off-the-shelf trained neural networks cannot be taken for granted, we will investigate architectures that are particularly suited for such approximations with an emphasis on so-called neural operators [23], as butterfly structures are more likely to appear naturally in this context.

Based on previous work on preserving the distribution of the weights, we will study sparse regularization and quantization approaches that promote the preservation of the distribution of the activations.

On a more practical side, we will study the efficient implementation of the proposed techniques and benchmark the developed approaches on targeted computer vision and NLP tasks.

#### 3.1.4 Memory Saving Techniques for Large Scale Model Training

**Resources:** One 2-year engineer starting 09/25

**Supervision:** hive (Alexandru Dobrila) and Topal (Olivier Beaumont, Lionel Eyraud-Dubois, Julia Gusak)

**Related Topics**: PhD Thesis 3.1.1

**Context:** The rapid development of LLMs trained on generic data has opened up spectacular capabilities in many fields [24, 25, 26]. It is common for users to wish to specialize an LLM for a particular purpose, and to carry out a fine-tuning phase prior to use [27]. This fine-tuning operation is much less expensive in terms of computation, but with the size of LLMs models, performing this operation requires a large amount of memory, and therefore recent GPUs. The aim here is to enable this fine-tuning operation to be carried out with less recent resources, but potentially in larger number, so that memory of resources can be aggregated to store the whole model. However, there is a significant communication cost associated with the use of distributed GPUs, and it may be appropriate to use fewer resources, by also aggregating CPU memory and precisely controlling the memory peak. Topal is developing re-materialization and offloading solutions that help to limit memory requirements during model training, with limited overhead in terms of execution time. In particular, OffMate [28] enables fine-tuning of large models such as Llama-3 on a single GPU.

**Goals:** The aim of this project is to make the OffMate open-source software stable and to port it to the hive Compute application stack. The work will involve professionalizing the development of this software, testing and validating it on the most popular LLM models, and maintaining documentation and tutorials to make it easier to use. A special effort will also be made to facilitate the deployment of OffMate in a hive Compute environment. The work will initially focus on training on a single GPU, but may be extended to enable training on multiple GPUs for larger models.

### 3.2 Axis 2: privacy and security for training, algorithmic, system and hardware approaches.

**Coordination: Davide Frey (Wide)**

### 3.2.1 Enhancing Privacy in Decentralized Machine Learning Through Homomorphic Encryption

**Resources:** PhD Thesis starting 09/25

**Supervision:** Wide (Davide Frey), Inria Rennes (Philippe Chartier), and Mathematics Institute of Rennes (Mohammed Lemou).

**Related Topics**: PhD Thesis 3.3.3, PhD Thesis 3.3.4.

**Context:** Decentralized machine learning presents a transformative approach, empowering users to train models locally and maintain control over their data [29]. However, this model introduces significant privacy risks, particularly through attacks such as membership inference, wherein an adversary may deduce whether certain data points were part of a user's training set. Such vulnerabilities underscore the pressing need for effective privacy-preserving mechanisms [30, 31].

**Goals:** The primary aim of this thesis is to investigate and implement strategies utilizing homomorphic encryption to address the privacy risks associated with decentralized model sharing. Key objectives include:

1. Identifying specific homomorphic encryption operations that can effectively prevent data leakage.
2. Exploring optimization techniques that navigate the computational limitations of homomorphic encryption, especially concerning the considerable costs associated with multiplicative operations.

Approach: The research will focus on several key areas:

— Mathematical Foundations: The PhD candidate will explore the theoretical aspects of homomorphic encryption and differential privacy, leveraging the expertise of Philippe Chartier and Mohammed Lemou, where they specialize in mathematical methods for homomorphic encryption [32]. This collaboration aims to integrate mathematical insights with practical computational applications.

— Distributed Algorithms and Machine Learning Structures: With Davide Frey's experience in the WIDE team at Inria, which specializes in distributed algorithms and decentralized machine learning [33, 34], the research will implement relevant algorithmic approaches. A selective encryption strategy will focus on encrypting only a subset of machine learning model parameters, seeking to balance privacy enhancement with computational efficiency. The validation of this hypothesis will involve both theoretical proofs and empirical investigations.

— Implementation Strategy: A significant focus will be on incorporating the newly developed privacy-preserving techniques into a decentralized machine-learning library, particularly within the framework of the FedMalin Inria challenge. This hands-on application will serve as a critical testing ground for assessing the effectiveness of the proposed methods.

### 3.2.2 Security and Enhancement for Next-gen Trusted RAG Yields using Confidential Computing

**Resources:** PhD Thesis starting 09/25

**Supervision:** Wide (David Bromberg and Davide Frey) for Inria and Alexandru Dobrila for Hive.

**Related Topics**: PhD Thesis 3.3.3, PhD Thesis 3.2.3

**Context:** Operating on distributed data, be it for inference or for training a distributed model holds the promise for more private forms of machine learning. Instead of having a single or a few service providers that collect user data, data can remain on user devices without being collected in large data silos. In spite of this, decentralized learning does not automatically guarantee privacy. Indeed, in a distributed setting any participant becomes a potential attacker. In this setting, the integration of confidential computing into artificial intelligence (AI) systems is becoming increasingly important [35] as organizations adopt AI for handling sensitive data in sectors like healthcare, finance, and defense. Retrieval-Augmented Generation (RAG) systems, which combine information retrieval with generative models, such as large language models (LLMs), are a popular solution for tasks like document summarization, chatbot development, and recommendation engines. However, they are often deployed in cloud environments, which introduces security risks, including data leakage, intellectual property (IP) theft, and model tampering.

The brand new Intel TDX CPU or AMD SEV-SNP features provide a promising solution by securing RAG systems through the isolation of both AI models and data within trusted execution environments (TEEs), even when deployed in multi-cloud or untrusted cloud infrastructures. Confidential computing features like TDX or SEV-SNP protect sensitive workloads by encrypting data during processing (data-in-use), while also ensuring model privacy and integrity using cryptographic attestation [36, 37].

**Goals:** This thesis plans to leverage server-side confidential computing devices to secure the processes of training and inference in decentralized machine learning. A number of approaches have already proposed the use trusted hardware in the context of federated learning [38, 39, 40]. But the use of TEEs in the context completely decentralized learning remains a niche application [41] particularly because many of these technologies like Intel's SGX are currently being discontinued. This calls for approaches that can integrate the novel generation of server-side trusted hardware (e.g. Intel TDX) with non-trusted devices.

In this PhD thesis, we aim to leverage the Wide team's expertise in Operating Systems and Hypervisors, on trusted execution environments [42], and decentralized machine learning [33, 34] to design a Trusted Virtual Machine Monitor (VMM) that will manage and secure Intel TDX-enabled Retrieval-Augmented Generation (RAG) systems, addressing key research objectives. Designing a Trusted VMM presents significant challenges due to the stringent requirements for security, performance, and scalability.

A first challenge lies in ensuring that the VMM operates transparently with the guest operating systems (OS). This entails providing essential security services—such as isolation and resource management—without modifying the guest OS or requiring it to be aware of the underlying TDX-aware VMM. Ensuring this transparency is critical for compatibility with existing systems and workloads, as modifying the guest OS would introduce substantial overhead and deployment complexity.

A second challenge results from the fact that not all available hardware will be equipped with confidential-computing devices. This results from the presence of non-trusted or incompatible server-side devices. To this end, we plan to leverage clustering approaches that can make it possible to leverage groups of devices that can communicate with each other. However, naive clustering would make the system very vulnerable to attacks. As a result, we plan to leverage a hybrid solution in which the data managed by a model is split between a protected and an unprotected part, leading to an inference or training process that is split between trusted and non trusted devices.

### 3.2.3 Middleware for the secure execution of AI jobs on heterogeneous consumer hardware architectures

**Resources:** PhD Thesis starting 09/25

**Supervision:** Stack (Guillaume Rosinosky and Mario Sudhölt) for Inria and Alexandru Dobrila for Hive.

**Related Topics**: PhD Thesis 3.3.3, PhD Thesis 3.3.4, PhD Thesis 3.2.2

**Context:** Taking security into account in distributed AI work is essential to respect user privacy, whether for inference or model training [43]. One way of overcoming this challenge is to employ the hardware security capabilities provided by modern processors, co-processors and chipsets [44, 45]. Security features present on modern consumer computers include, for example, the Trusted Platform Management (TPM) chipsets integrated into the majority of recent x86 motherboards, or Microsoft Pluton [46] in modern x86 processors. These modules provide numerous functions to help secure applications, such as key storage, encryption and decryption. Added to this are the capabilities of ARM processors to securely run code in isolation, using TrustZone technology [47]. These have been successfully employed on AI projects in recent work [45, 48], and this type of platform (ARM) is becoming increasingly widespread in consumer computers and data centers. However, not all these capabilities are identical, nor do they provide the same security guarantees depending on the family and model of processors and chipsets employed. Added to this is the complexity of using this type of architecture, which often requires specific programming models resulting in heavy adaptations of the executor code [48], or in other cases a very reduced performance [43].

**Goals:** The main objective of this PhD is to provide middleware-like software support able to distribute and run AI workloads on resources hosted on end-user computers, according to their software and hardware capabilities, while taking into account users' privacy requirements. Several research questions will be considered:

— how to ensure the security of AI processing on computers with heterogeneous capabilities, taking advantage of TPM and TrustZone?

— how to adapt AI processing to take advantage of these security capabilities?

— what trade-offs between security and performance need to be addressed, and how can jobs be dynamically distributed across participating nodes?

The development of this middleware is based on the following steps: First, an analysis of the capabilities and limitations of TPM and TrustZone will be carried out to understand how these technologies can secure AI processing. Then, the middleware will be designed to automatically detect the hardware capabilities of computers and adjust security level according to the available configuration. To achieve this goal, it will integrate mechanisms to isolate AI processing with TrustZone and secure the storage of sensitive data with TPM. Finally, an AI job orchestrator based on optimization models and/or heuristics to determine the most efficient deployment according to security, performance and machine availability needs. Experiments on devices equipped with these technologies will validate the middleware. Tests will measure the impact of dynamic adaptation on security and performance, providing data on the feasibility and effectiveness of the middleware in real-world scenarios.

### 3.2.4 Confidential containers for AI worklads

**Resources:** One 2-year engineer starting 09/26

**Supervision:** hive (Alexandru Dobrila), Stack (Guillaume Rosinosky) and Wide (Davide Frey)

**Related Topics**: PhD Thesis 3.2.2, PhD Thesis 3.2.3.

**Context:** The increasing use of AI applications on user devices (PCs, laptops) raises concerns about the protection of sensitive data. The migration of AI workloads to local platforms outside the cloud increases the need for robust security solutions on end-user devices. Some processors offer code execution isolation features, such as Intel TDX [49] and ARM TrustZone [47]. In addition to these, chipsets such as Trusted Platform Module (TPM) and Microsoft Pluton [44] offer advanced security mechanisms that are still under-exploited in user environments. The combination of confidential containers and these technologies would make it possible to guarantee a secure environment for AI, directly on PCs and personal computers. Existing initiatives such as Confidential Containers [50] are generalist and limited in the performance they offer [43].

**Goals:** The aim of this research is to design a customized confidential container model for AI workloads on user devices, integrating advanced security technologies such as TDX, TrustZone and TPM:

- Development of a secure container for user PCs: Create a prototype container capable of isolating sensitive AI data and running efficiently on personal machines, while guaranteeing a high level of privacy.
- Integration of client-side TrustZone and TPM technologies with their datacenter-side counterpart TDX: Ensure enhanced security through a hybrid architecture combining TDX for Intel processors, TrustZone for ARM architectures, and TPM for key management and authentication.
- Security and performance evaluation: Test the effectiveness of the solution in terms of data isolation, resistance to advanced threats and performance for AI workloads, validating its suitability for uses requiring a high degree of privacy.

A prototype executor will also be made available to the community as part of the reproducible research.

This research proposes to design and evaluate a confidential container prototype for PCs, taking advantage of TDX for secure isolation on Intel processors, TrustZone for runtime protection on ARM, and TPM for key management and authentication. In the state of the art, current approaches to secure containers often extend system calls, use Library OS or are based on WebAssembly [48]. The extension of approaches such as Confidential Containers [51], Google gVisor [52, 53] or Open Enclave SDK [54] can be considered for this work. The prototype will be tested in a variety of environments to measure its effectiveness in terms of privacy protection and performance. The approach will focus on a flexible architecture compatible with different types of hardware to meet heterogeneous user needs.

### 3.2.5 Distributed training with Malicious clients

**Resources:** PhD Thesis starting 09/26

**Supervision:** Coati (Chuan Xu) and Neo (Samir Perlaza) for Inria and Igor Carrara for Hive.

**Related Topics**: PhD Thesis 3.3.1

**Context:** Federated Learning (FL) empowers a multitude of IoT devices, including mobile phones and sensors, to collaboratively train a global machine learning model while retaining their data locally [55, 56]. A prominent example of FL in action is Google's Gboard, which uses a FL-trained model to predict subsequent user inputs on smartphones [57]. However, in the absence of protective measures, malicious participants in FL can easily derail the learning process or even extract sensitive information from other participants. For example, a malicious agent can deteriorate the model performance by simply flipping the labels [58] and/or the sign of the gradient [59] and even inject backdoors into the model [60] (backdoors are hidden vulnerabilities, which can be exploited under

certain conditions predefined by the attacker, like some specific inputs). Furthermore, a malicious participant can infer when a particular property appears in another participant's training dataset by locally optimizing an alternate loss function [61]. Additionally, it can also infer class representations of a target participant by secretly training a local generator and submitting a manipulated model update [62].

**Goals:** The objective of this PhD research is to investigate privacy vulnerabilities in Federated Learning (FL) systems and in distributed training systems more in general, particularly when malicious participants poison local models. First, we aim to advance existing reconstruction attacks to extract private information beyond simple class representations. These attacks are intended to operate stealthily, making them difficult to detect using conventional intrusion detection methods. Second, we seek to develop a theoretical framework to quantify privacy leakage within a bounded model-poisoning neighborhood. The bounded nature of the attack ensures that it remains undetectable to some extent - an area that remains largely unexplored in current research. Finally, our goal is to identify or design the most effective defenses to mitigate these attacks while maintaining the high performance of the final model.

This research will integrate theoretical analysis with empirical validation to investigate privacy vulnerabilities in FL systems. We will develop advanced model poisoning attacks inspired by geometric properties to enhance existing model poisoning techniques for privacy, enabling greater information leakage about honest participants. A theoretical framework will be constructed that defines key privacy metrics, such as mutual information and KL divergence, while incorporating the concept of bounded neighborhoods to effectively quantify privacy leakage, building on the literature [63, 64]. Empirical validation will occur within a controlled simulation environment, using relevant datasets to assess both the performance of the model and the effectiveness of implemented defense mechanisms.

## 3.3 Axis 3: Large-scale computing on intermittent resources

**Coordination: Giovanni Neglia (Neo)**

### 3.3.1 Management of complex applications on volatile heterogeneous platforms

**Resources:** PhD Thesis starting 09/26

**Supervision:** Coast (Thomas Lambert, Claudia-Lavinia Ignat) and Magellan (Shadi Ibrahim) for Inria and Mamoutou Diarra for Hive.

**Related Topics**: PostDoc 3.3.2

**Context:** We are interested in studying how to facilitate and optimise the execution of batch jobs (e.g., MapReduce applications, data-intensive applications) in hive Net-like environments. The challenges we face in this context are: (i) the heterogeneity of the platform in terms of compute, memory and network, (ii) resource dynamicity and (iii) node availability (churns).

**Goals:** The aim of this task is to provide reliable and scalable data processing on large-scale, trusted P2P systems. This research is expected to make innovative contributions in the following aspects:

— We first plan to study static allocation strategies from the perspective of P2P systems where, unlike clouds, resources are extremely heterogeneous. Designing a practical, realistic yet tractable

model will be one of our first objectives.

— To cope with the dynamic nature of resources, allocation decisions should be made during execution - at runtime. The aim is to design a scheduling policy that adapts to resource dynamicity, even in the absence of up-to-date information about the whole P2P system. The balance between static and dynamic approaches (and its potential, as demonstrated in earlier work [65]) will be an important aspect of this work.

— We will also work on a scheduling policy when multiple tasks (that belong to different jobs) share part of their input files. For example, in the hive Net Private context, we can consider that the dataset owned by the company can be used for different purposes at the same time. Similar problems have already been addressed for compute-intensive tasks and in HPC systems [66, 67], but not for data-intensive applications in highly distributed environments.

— Finally, resource dynamicity lead to performance variability and thus to stragglers (slow tasks). This prolongs the execution of applications as the execution time depends on the completion time of these tasks [68]. We plan to investigate how to detect stragglers in such heterogeneous environments and how to deal with them efficiently - at runtime, by adapting techniques such as cloning and speculative execution [68, 69].

### 3.3.2 Fault Tolerance in hive Compute

**Resources:** One 1-year PostDoc starting 09/2026

**Supervision:** Magellan (Shadi Ibrahim) for Inria and Mamoutou Diarra for Hive

**Related Topics**: PhD Thesis 3.3.1, PhD Thesis 3.1.2

**Context:** Unlike clouds, large-scale P2P environments are characterized by a high number of node failures and churns. This can lead to unwanted delays in the completion time of running applications and makes both scalability and reliability critical when running data-intensive applications (e.g., MapReduce applications) in a hive Compute environment. We are interested in investigating how to optimize the execution of data-intensive applications in the presence of failures and churns by leveraging the hive-Disk platform, using checkpoints and making job scheduling failure-aware.

**Goals:** General purpose fault tolerant strategies lead to excessive execution of recovery tasks (re-execution of tasks on failed machines). Therefore, we will investigate how to adapt fault-tolerance techniques to P2P systems by making job scheduling failure-aware (leveraging our previous experience and work with Hadoop clusters [70, 71]) and by enabling checkpoint/restart so that we can roll back execution from the last checkpoint instead of restarting the execution after a failure [72]. We will present a performance model for checkpoint/restart in P2P systems and introduce a scheduling framework that decides when and where to trigger checkpoints and where to restart, and when and where to execute recovery tasks, taking into account failure distribution, data location, and resource heterogeneity. We will also explore how the hive-Disk platform can be used to store checkpoints and temporary data (e.g., map outputs in MapReduce).

### 3.3.3 Inference and Training in a Large-Scale Volatile environment

**Resources:** PhD Thesis starting 09/2026

**Supervision:** Mimove (Nikolaos Georgantas) and Wide (Davide Frey) for Inria and Igor Carrara for Hive.

**Related Topics**: PhD Thesis 3.1.1, PhD Thesis 3.1.2, PhD Thesis 3.2.5

**Context:** hive aims at deploying a large-scale inference and training engine on user machines distributed across the planet. This poses significant challenges due to the open and heterogeneous nature of such a network environment, which may include resource-constrained nodes that can join and leave at any time. In this context, two research directions can be identified.

Firstly, inference and even more training are applications that are very demanding in terms of resources. We aim to enable effective inference and training in such an environment, while addressing the dynamicity, resource constraints and distribution of the participating devices.

Secondly, nodes may operate maliciously [73] and they may carry out so-called sybil attacks, with a malicious user's device impersonating a large number of machines. Addressing these attacks in the context of decentralized machine learning poses different challenges with respect to a general-purpose distributed system. On the one hand, the inherent redundancy associated with machine learning models may provide some partial protection from malicious behavior that can be complemented with classical approaches. On the other, the complex architecture of machine learning models calls for replication techniques that concentrate on the most critical components of a model.

**Goals:** The goal of this thesis lies in addressing the challenges associated with decentralized machine learning in the context of large-scale systems. We will consider the two main settings of inference, and training in a distributed setting.

Regarding the first identified research direction, we will address optimal scheduling of machine learning models over a network of heterogeneous, volatile and resource constrained nodes. We will aim at optimizing metrics such as resource consumption or latency and deal in an adaptive manner with the dynamic resource network [74]. We will rely on existing model architectures and techniques of distribution, parallelism, compression and acceleration for resource constrained devices [75], while taking into account large-scale communication between nodes. While addressing inference in a first step, we will extend our study to include training, where, additionally, distributed model synchronization should be taken into account.

Regarding the second research direction and in the context of inference, the first challenge consists of addressing the malicious and erratic behavior of participating nodes. This can include nodes that purposely try to disrupt the inference process, and nodes that experience bugs or other kinds of failures, collectively known as Byzantine failures [73], or simply nodes that disconnect permanently or temporarily causing churn [76]. The second challenge consists of addressing Sybil attacks in addition to byzantine behavior [77]. This turns out to be particularly difficult as the attacker's ability to create an arbitrary number of identities prevents the use of deterministic algorithms that rely on the knowledge of the number of network participants. In the context of training, the challenges of Byzantine behavior and Sybil attacks remain and will require novel techniques to design resilient training algorithms.

More specifically, regarding the first research direction, we will rely on our previous research results in the MIMOVE team on optimal placement of data stream operators in the edge-fog-cloud continuum [78, 79]. We will devise new efficient and adaptive algorithms that take into account the particularities of inference and training in the demanding large-scale volatile environment in question.

Regarding the second research direction, we plan to start by studying how existing models can be split and replicated during the inference process. In particular, we plan to identify for each considered model, the portions of the model that are most sensitive to byzantine behavior and focus on approaches

that replicate these parts of the models. This will make it possible to define novel algorithms that can efficiently tackle byzantine behavior in the context of a distributed inference process. In a second step, we plan to address Sybil attacks by combining the techniques recently developed by the WIDE team [80] with novel approaches based on the design of an identity management system for devices. We also plan to address the issue of training a model in a large-scale distributed environment. In this case, we will consider both model and data parallelism [81]. In the case of model parallelism, we may be able to extend some of the techniques developed for the inference process to the case of training. Finally, to address the volatile and dynamic nature of large-scale networks, we will leverage the recent results of the WIDE team on the topic and design novel probabilistic algorithms that can support both inference and training.

### 3.3.4 Frugal Distributed Training with Volatile Resources

**Resources:** PhD Thesis starting 09/2026

**Supervision:** Argo (Kevin Scaman) and Neo (Giovanni Neglia) for Inria and Igor Carrara for Hive.

**Related Topics**: PhD Thesis 3.1.1, PhD Thesis 3.3.3.

**Context:** The hive distributed platform allows participants to share computing and storage resources for limited periods. While resource availability may sometimes be known in advance, it is also prone to unpredictable fluctuations, with resources being withdrawn unexpectedly. This challenge is central to cross-device federated learning (FL) systems, where client participation is often dependent on factors like battery levels and WiFi connectivity. In operational FL systems, this volatility is typically managed by recruiting a large pool of clients at each round and proceeding once a sufficient number of responses are received. However, this approach can introduce biases, which have been addressed in the literature only for basic stochastic [82, 83, 84] or cyclic participation models [85, 86], the latter being motivated by time-of-day patterns in client availability.

**Goals:** The hive use case presents several distinctive challenges. First, the system operates on a much smaller scale, and concerns about efficiency and cost-effectiveness limit the feasibility of recruiting large numbers of participants to ensure redundancy. Second, while FL systems must contend with the inability to move datasets across clients (and the consequent difficulties due to statistical heterogeneity), hive may permit the transfer of portions of data to participants. Lastly, hive clients' availability may be known in advance or at least predicted with greater accuracy. The objective of this PhD is to develop new distributed training algorithms tailored for the hive environment. These algorithms will determine which resources to utilize, when to engage them, and how they should be employed—for example, how much data to transfer and how many consecutive model updates a client should perform before communicating back.

This research will involve several key phases, requiring different methodologies. First, a modeling phase will focus on characterizing the availability patterns of hive participants using real-world traces or insights from hive engineers. The next phase will involve algorithm design, with the aim of developing solutions that come with theoretical guarantees regarding training time duration. These algorithms will initially be tested in a simulated environment using standard machine learning datasets, before being deployed and evaluated in the hive system.

### 3.3.5 Transient heterogeneous clusters for Big Data and AI workloads

**Resources:** One 2-year engineer starting 09/2026

**Supervision:** Stack (Guillaume Rosinosky) and Igor Carrara for Hive.

**Related Topics**: PhD Thesis 3.3.1

**Context:** The operation of IT infrastructures, whether personal or professional, remains largely under-optimized, resulting in excessive energy consumption with no real added value. To overcome these inefficiencies, technologies such as virtualization, containerization and Function as a Service (FaaS) solutions have enabled developers to move away from dependences on a specific hardware platform [87]. In this context, Kubernetes is a solution for managing workloads at a scale that is ubiquitous not only in data centers, but also right up to the network edges (KubeEdge, K3S) [88]. This is particularly the case for resource-intensive Big Data and AI workloads. However, the use of Kubernetes on end-user workstations remains marginal, apart from rare initiatives such as Akash Network [89]. Major challenges to address include the need to take into account the heterogeneity of platforms and their characteristics, and to consider transient work nodes that can be started or stopped by users when required [90].

**Goals:** The aim of this work is to study the effects of using Kubernetes worker nodes on different operating systems, in particular Windows and Linux for running Big Data and AI jobs on fleets of user computers. As users can provision and release machines at any time, an in-depth performance analysis will be carried out, including the study of "cold start" phenomenon caused by the execution of containers and virtual machines. Finally, the impact of releasing machines will be measured in relation to the user's workload, using metrics such as processor consumption, memory usage, disk I/O and network bandwidth.

A prototype executor will also be made available to the community as part of the reproducible research. The proposed approach includes a state-of-the-art analysis, followed by the development of an executor prototype based on well-established tools such as WSL, K3S or KubeEdge. This prototype will be used to run Big Data workloads (such as Apache Flink or Spark) and AI workloads, based on benchmarks recognized by the scientific and industrial communities. A detailed analysis will be made of the time required to deploy a new cluster, install and release compute containers, depending on the capabilities of the target machine, the framework (Big Data, AI), the Kubernetes distribution (K3S, KubeEdge, etc.) and the Operating System (Windows, Linux).

# 4 PROJECT MANAGEMENT

## 4.1 Governance

As per the existing partnership agreement, the project will be directed through :

1 Steering Committee composed of :

- Inria: The Deputy Director General for Innovation (François Cuny or his representative) + the Deputy Director General for Science (Jean Frédéric Gerbeau or his representative)
- the Partner: Antoine Clerget + David Gurlé - CEO and executive directors of hive

1 Management Committee composed of :

- Inria: 1 Scientific Director (Olivier Beaumont) + 1 Operational Director (Amar Bouali)

— the Partner: Alexandru Dobrila - hive Technology Director

In addition, each axis has a scientific coordinator, who is responsible for the animation of the axis, preparing activity syntheses for plenary meetings, and organizing regular videoconference meetings within each axis.

The Steering Committee will meet once a year, with representatives of the Management Committee. Its role is to make all strategic level decisions aimed at ensuring the harmonious development of the partnership/Joint Laboratory, in accordance with the objectives, priorities and interests of both Parties. Its members have the authority to commit the institutions to the decisions made.

The Management Committee will meet every 8-10 weeks (at least twice a year), with various members of the project. The Management Committee is responsible for implementing the Steering Committee's strategy, monitoring and ensuring the day-to-day coordination of the Partnership and its work, drawing up consolidated activity reports, ensuring relations with third parties and, more generally, for the research topics covered by the Partnership.

More details are available in the existing partnership agreement.

In addition, to ensure good synchronisation and coordination with the various national projects on related
topics, representatives will be appointed with PEPR Secure Compute https://www.pepr-cybersecurite.fr/projet/securecompute/, PEPR IA https://www.pepr-ia.fr, PEPR Cloud https://pepr-cloud.fr/fr/, PEPR Numpex https://anr.fr/fr/france-2030/programmes-et-equipements-prioritaires-de-recherche-pepr/numerique-pour-lexascale-numpex/ and the Inria Challenge Fed-Malin https://project.inria.fr/fedmalin/

## 4.2 Resources

In order to achieve the objectives of this challenge we need the following resources:

— 9 PhD positions distributed over the three axis (2 in Axis 1 on Frugality, 4 in Axis 2 on Security and Privacy, 3 in Axis 3 on Volatility)

— 2 Postdoc positions (3 years in total) in Axis 1 and Axis 3.

— 3 Engineer positions (6 years in total) who will prototype the results of the solutions proposed within the PhD theses or build an environment to help the implementation of PhD works.

— 12 Internship positions (6 years in total) associated to PhD and Engineering positions.

A joint operational budget will cover expense needs for the product, typically including travel costs such as:

— two plenary meetings/workshops per year with all the project members;

— two travels of two weeks per year for each PhD student in the research center that ensures the co-supervision of the thesis;

— one travel of one week per year for each permanent member that supervises/co-supervises a thesis in the research center that supervises/co-supervises the respective thesis. This travel will be synchronised with the travel of the Ph.D. student;

— two travels of one week per year for each engineer/postdoc to the partners involved in the task.

— One mission per year and per PhD student or Postdoc to present the results of Cupseli at international conferences.

## 4.3 Dissemination and visibility.

Dissemination and transfer of knowledge are both internal and external activities to the project consortium. Within the former, it is a process of improving knowledge among partners. As for external dissemination, it will be focused on scientific communication.

All consortium partners will contribute to the dissemination activities by means of:

- writing of research papers and participation to peer-reviewed international conferences and journals.
- producing and publishing open-source code implementations;
- setting-up and maintenance of various dissemination tools, e.g. web-site;
- organizing and participating in the internal workshops where outcomes of the project will be demonstrated.

All documentations are expected to be public. They will first circulate inside the project and will be made public as soon as the involved partners have declared their consent.

An Internet website will be developed from the very start of the project, whose main objective is to diffuse the challenge objectives and results as wide as possible, throughout the community and over.

## 4.4 Future possibilities (at the end of the Challenge)

The collaboration with hive will open up new research topics for Inria teams from future business use cases of hive like AI training on higly distributed and heterogeneous ressources for instance.

The distributed compute based on available ressources from the community proposed in this challenge contribute to the overall target of carbon foot print reduction the tech industry is facing. This project ultimately represents a greener solution for the upcoming AI challenges by provide another way to address the growing power requirments than building new Dataceeters. As such we intend to apply for national and internatinoal call for project like PIA IV, i-démo or Horizon Europe.

Moreover, through the Security and Privacy axis (Section 2.1.2), the project might also contribute to the GAIA-X initiative (https://www.data-infrastructure.eu) that proposes the next generation of open, transparent and secure data infrastructure where data and services are shared in a trusted environment.

# 5 IMPACT AND RESULTS EXPLOITATION

Users are increasingly concerned about the security and privacy of their data, the dependence on centralized cloud service providers for data storage and for computations, and the growing impact of data centers on our planet. The aim of hive is to address these concerns and give users total control over their data and computation in a secure, private environment.

## 5.1 Expected impact

To achieve the objectives and meet the expectation mentioned above, hive needs to establish a research culture that will play an active role in the development of the hive technology through an initiative geared towards generating innovation tailored to its specific needs.

This joint challenge between hive and Inria is considered as a "strategic partnership", set up specifically to strengthen the innovative capacity of hive and to create a permanent link between hive engineers and Inria researchers.

This partnership will let Inria and hive share their expertise – namely, hive's technical skills and Inria's scientific skills in the digital domain. The aim is to advance the distributed cloud assets for the benefit of cloud users, by overcoming the scientific and technical obstacles currently standing in hive's way. This strategic partnership initiative provides a way of answering the challenges discussed in Section 1.

In the very short term, this strategic partnerships will create job opportunities in France by hiring nine PhD students, three research engineers, two postdocs, and a few interns.

In the mid/long terms, hive intends to hire the graduated PhD students and the interns as permanent research engineers and continue the collaboration with Inria while hiring other PhD students to address new challenges eventually.

## 5.2 Sharing and results exploitation

The results of this partnership may be of diverse nature and consist of both tangible results as well as of skills and personal experiences that both project organizers and participants to the activities have acquired.

Tangible results may include for example:

- approaches, models or algorithm to solve a specific problem;
- a practical tool or a software;
- research reports or studies (e.g. scientific publication).

Public disclosure of some project key results by any appropriate means is very important for both hive and Inria. It makes research results known to various stakeholder groups (e.g., research peers, industry and other commercial actors, professional organisations, policymakers) in a targeted way, enabling them to use the results in their own work. This process will be planned and organized at the beginning of each sub-topic of this joint collaboration. The results could be shared through the following means:

- scientific publications;
- contribution to open-source code;
- events: exhibitions, workshops, demo days, cluster events;
- participation to third-party events: scientific conferences;
- other dissemination supports: public website, press releases, white papers, online tools and training for specific target groups.

Alternatively, some joint key results may constitute a patentable invention and may be deemed by the Steering Committee to warrant the filing of a Patent shall be patented in the joint names of Inria and Hive.

During the first months of the partnership, the steering committee will define the dissemination and exploitation strategies focusing on the planned project outcomes and targeted stakeholders. This will be updated each year after annual monitoring. The planning and execution of the project dissemination activities require a schedule closely aligned with key collaboration objectives and milestones. At this scope, the project will be organized around 3 phases:

- Initial awareness phase (Month 0-6) to ensure the partnership is known to relevant stakeholders

and the public in general. In this phase, we will communicate through digital medias and through the Inria and hive websites to announce the partnership and share its main objectives.

— Targeted dissemination phase (Month 6-30): attend selected events. Preliminary project results will be presented to the target audiences through participation in events and scientific conferences, scientific publications, organisation of workshops, creation of communication materials, media general outreach through press releases and articles in magazines.

— Presentation of results (Month 30-36): this represents the period when the project reaches its most significant outputs. This will be the more active period in the whole dissemination strategy, matching with the publications and the defense of the PhD theses. Exploitation of these results will also be ensured by outlining the actions required to fulfill their market potential.

The main focus for the researchers, engineers and PhD students involved in this collaboration will be on the development of methods, models, algorithm and prototypes around the main challenges and topics that we have identified. These results will be progressively leveraged by the hive R&D team by integrating them into the hive solution.

## REFERENCES


[1] Jananie Jarachanthan, Li Chen, Fei Xu, and Bo Li. Amps-inf: Automatic model partitioning for serverless inference with cost efficiency. In *Proceedings of the 50th International Conference on Parallel Processing*, 2021.

[2] Olivier Beaumont, Jean-François David, Lionel Eyraud-Dubois, and Samuel Thibault. Exploiting processor heterogeneity to improve throughput and reduce latency for deep neural network inference. In *SBAC-PAD 2024-IEEE 36th International Symposium on Computer Architecture and High Performance Computing*, 2024.

[3] Anne Benoit. *Scheduling Pipelined Applications: Models, Algorithms and Complexity*. Habilitation à diriger des recherches, Ecole normale supérieure de lyon - ENS LYON, July 2009.

[4] Thomas Heinze, Leonardo Aniello, Leonardo Querzoni, and Zbigniew Jerzak. Cloud-based data stream processing. In *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*, DEBS '14, page 238–245, New York, NY, USA, 2014. Association for Computing Machinery.

[5] Haruna Isah, Tariq Abughofa, Sazia Mahfuz, Dharmitha Ajerla, Farhana Zulkernine, and Shahzad Khan. A survey of distributed data stream processing frameworks. *IEEE Access*, 7:154300–154316, 2019.

[6] N. T. Karonis, B. de Supinski, I. Foster, W. Gropp, and E. Lusk. A multilevel approach to topology-aware collective operations in computational grids, 2002.

[7] Leah Shalev, Hani Ayoub, Nafea Bshara, and Erez Sabbag. A cloud-optimized transport protocol for elastic and scalable hpc. *IEEE Micro*, 40(6):67–73, 2020.

[8] Gonzalo Martín, Maria-Cristina Marinescu, David E. Singh, and Jesús Carretero. FLEX-MPI: an MPI extension for supporting dynamic load balancing on heterogeneous non-dedicated systems. In *Proceedings of the 19th International Conference on Parallel Processing*, Euro-Par'13, page 138–149, Berlin, Heidelberg, 2013. Springer-Verlag.

[9] Jonathan Frankle and Michael Carbin. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *International Conference on Learning Representations*, September 2018.

[10] Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing Lottery Tickets: Zeros, Signs, and the Supermask. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019 (NIPS 2019)*, page 3592–3602, 2019.

[11] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart Van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*, 2021.

[12] Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[13] Antoine Gonon. *Harnessing symmetries for modern deep learning challenges : a path-lifting perspective*. Theses, Ecole normale supérieure de lyon - ENS LYON, November 2024.

[14] Quoc-Tung Le. *Algorithmic and theoretical aspects of sparse deep neural networks*. Theses, Ecole normale supérieure de lyon - ENS LYON, December 2023.

[15] Arthur da Cunha, Francesco D'Amore, and Emanuele Natale. Polynomially Over-Parameterized Convolutional Neural Networks Contain Structured Strong Winning Lottery Tickets. In *Thirty-Seventh Conference on Neural Information Processing Systems*, November 2023.

[16] Emanuele Natale, Davide Ferre', Giordano Giambartolomei, Frédéric Giroire, and Frederik Mallmann-Trenn. On the Sparsity of the Strong Lottery Ticket Hypothesis. In *38th Conference on Neural Information Processing Systems (NeurIPS 2024)*, Vancouver, Canada, December 2024.

[17] T. Dao, B. Chen, N. S. Sohoni, A. D. Desai, M. Poli, J. Grogan, A. Liu, A. Rao, A. Rudra, and C. Ré. Monarch: Expressive structured matrices for efficient and accurate training. In *International Conference on Machine Learning*, pages 4690–4721. PMLR, 2022.

[18] T. Dao, A. Gu, M. Eichhorn, A. Rudra, and C. Ré. Learning fast algorithms for linear transforms using butterfly factorizations. In *International Conference on Machine Learning*, pages 1517–1527. PMLR, 2019.

[19] Quoc-Tung Le, Léon Zheng, Elisa Riccietti, and Rémi Gribonval. Fast learning of fast transforms, with guarantees. In *ICASSP 2022 - IEEE International Conference on Acoustics, Speech and Signal Processing*, Singapore, Singapore, May 2022. This paper is associated to code for reproducible research available at https://hal.inria.fr/hal-03552956.

[20] Léon Zheng, Elisa Riccietti, and Rémi Gribonval. Efficient Identification of Butterfly Sparse Matrix Factorizations. *SIAM Journal on Mathematics of Data Science*, 5(1):22–49, 2023.

[21] R. Gribonval, T. Mary, and E. Riccietti. Optimal quantization of rank-one matrices in floating-point arithmetic—with applications to butterfly factorizations. Preprint, Under review, June 2023.

[22] R. Gribonval, T. Mary, and E. Riccietti. Scaling is all you need: quantization of butterfly matrix products via optimal rank-one quantization. In *29ème Colloque sur le traitement du signal et des images (GRETSI)*, number 2023-1193 in Actes du GRETSI 2023, pages 497–500, Grenoble, France, August 2023. GRETSI - Groupe de Recherche en Traitement du Signal et des Images.

[23] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural Operator: Learning Maps Between Function Spaces, May 2024. arXiv:2108.08481.

[24] Quzhe Huang, Mingxu Tao, Zhenwei An, Chen Zhang, Cong Jiang, Zhibin Chen, Zirui Wu, and Yansong Feng. Lawyer llama technical report. *arXiv preprint arXiv:2305.15062*, 2023.

[25] Chaoyi Wu, Xiaoman Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. Pmc-llama: Further finetuning llama on medical papers. *arXiv preprint arXiv:2304.14454*, 2023.

[26] Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. Large language models in finance: A survey. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, pages 374–382, 2023.

[27] Xin Luna Dong, Seungwhan Moon, Yifan Ethan Xu, Kshitiz Malik, and Zhou Yu. Towards next-generation intelligent assistants leveraging llm techniques. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5792–5793, 2023.

[28] Xunyi Zhao, Lionel Eyraud-Dubois, Théotime Le Hellard, Julia Gusak, and Olivier Beaumont. OFFMATE: full fine-tuning of LLMs on a single GPU by re-materialization and offloading. working paper or preprint, July 2024.

[29] Enrique Tomás Martínez Beltrán, Mario Quiles Pérez, Pedro Miguel Sánchez Sánchez, Sergio López Bernal, Gérôme Bovet, Manuel Gil Pérez, Gregorio Martínez Pérez, and Alberto Huertas Celdrán. Decentralized federated learning: Fundamentals, state of the art, frameworks, trends, and challenges. *IEEE Communications Surveys & Tutorials*, 25(4):2983–3013, 2023.

[30] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models . In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, Los Alamitos, CA, USA, May 2017. IEEE Computer Society.

[31] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914, 2022.

[32] Philippe Chartier, Michel Koskas, Mohammed Lemou, and Florian Méhats. Fully homomorphic encryption on large integers. Cryptology ePrint Archive, Paper 2024/155, 2024.

[33] Amaury Bouchra Pilet, Davide Frey, and Francois Taïani. Simple, efficient and convenient decentralized multi-task learning for neural networks. In Pedro Henriques Abreu, Pedro Pereira Rodrigues, Alberto Fernández, and João Gama, editors, *Advances in Intelligent Data Analysis XIX*, pages 37–49, Cham, 2021. Springer International Publishing.

[34] Sayan Biswas, Davide Frey, Romaric Gaudel, Anne-Marie Kermarrec, Dimitri Lerévérend, Rafael Pires, Rishi Sharma, and François Taïani. Low-cost privacy-aware decentralized learning, 2024.

[35] Kapil Vaswani, Stavros Volos, Cedric Fournet, Antonio Nino Diaz, Ken Gordon, Balaji Vembu, Sam Webster, David Chisnall, Saurabh Kulkarni, Graham Cunningham, Richard Osborne, and Daniel Wilkinson. Confidential computing within an AI accelerator. In *2023 USENIX Annual Technical Conference (USENIX ATC 23)*, pages 501–518, Boston, MA, July 2023. USENIX Association.

[36] Xinyang Ge, Hsuan-Chi Kuo, and Weidong Cui. Hecate: Lifting and shifting on-premises workloads to an untrusted cloud. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 1231–1242, New York, NY, USA, 2022. Association for Computing Machinery.

[37] Ziqiao Zhou, Yizhou Shan, Weidong Cui, Xinyang Ge, Marcus Peinado, and Andrew Baumann. Core slicing: closing the gap between leaky confidential VMs and bare-metal cloud. In *17th USENIX Symposium on Operating Systems Design and Implementation (OSDI 23)*, pages 247–267, Boston, MA, July 2023. USENIX Association.

[38] Akshay Gangal, Mengmei Ye, and Sheng Wei. Hybridtee: Secure mobile dnn execution using hybrid trusted execution environment. In *2020 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*, pages 1–6, 2020.

[39] Fan Mo, Hamed Haddadi, Kleomenis Katevas, Eduard Marin, Diego Perino, and Nicolas Kourtellis. Ppfl: privacy-preserving federated learning with trusted execution environments. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '21, page 94–108, New York, NY, USA, 2021. Association for Computing Machinery.

[40] Aghiles Ait Messaoud, Sonia Ben Mokhtar, Vlad Nitu, and Valerio Schiavoni. Shielding federated learning systems against inference attacks with arm trustzone. In *Proceedings of the 23rd*

*ACM/IFIP International Middleware Conference*, Middleware '22, page 335–348, New York, NY, USA, 2022. Association for Computing Machinery.

[41] Akash Dhasade, Nevena Dresevic, Anne-Marie Kermarrec, and Rafael Pires. TEE-based decentralized recommender systems: The raw data sharing redemption . In *2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 447–458, Los Alamitos, CA, USA, June 2022. IEEE Computer Society.

[42] Matthieu Pigaglio, Joachim Bruneau-Queyreix, Yerom-David Bromberg, Davide Frey, Etienne Riviere, and Laurent Reveillere. RAPTEE: Leveraging trusted execution environments for Byzantine-tolerant peer sampling services . In *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*, pages 603–613, Los Alamitos, CA, USA, July 2022. IEEE Computer Society.

[43] Fan Mo, Zahra Tarkhani, and Hamed Haddadi. Machine Learning with Confidential Computing: A Systematization of Knowledge. *ACM Comput. Surv.*, 56(11):281:1–281:40, June 2024.

[44] Lianying Zhao, He Shuang, Shengjie Xu, Wei Huang, Rongzhen Cui, Pushkar Bettadpur, and David Lie. A Survey of Hardware Improvements to Secure Program Execution. *ACM Computing Surveys*, 56(12):1–37, December 2024.

[45] Aghiles Ait Messaoud, Sonia Ben Mokhtar, Vlad Nitu, and Valerio Schiavoni. Shielding federated learning systems against inference attacks with ARM TrustZone. In *Proceedings of the 23rd ACM/IFIP International Middleware Conference*, Middleware '22, pages 335–348, New York, NY, USA, November 2022. Association for Computing Machinery.

[46] Vinay Pamnani. Microsoft Pluton security processor, July 2024.

[47] Qinyu Zhu, Quan Chen, Yichen Liu, Zahid Akhtar, and Kamran Siddique. Investigating TrustZone: A Comprehensive Analysis. *Security and Communication Networks*, 2023(1):7369634, 2023. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2023/7369634.

[48] Arttu Paju, Muhammad Owais Javed, Juha Nurmi, Juha Savimäki, Brian McGillion, and Billy Bob Brumley. SoK: A Systematic Review of TEE Usage for Developing Trusted Applications. In *Proceedings of the 18th International Conference on Availability, Reliability and Security*, ARES '23, pages 1–15, New York, NY, USA, August 2023. Association for Computing Machinery.

[49] Pau-Chen Cheng, Wojciech Ozga, Enriquillo Valdez, Salman Ahmed, Zhongshu Gu, Hani Jamjoom, Hubertus Franke, and James Bottomley. Intel TDX Demystified: A Top-Down Approach. *ACM Computing Surveys*, 56(9):1–33, October 2024.

[50] Confidential Containers.

[51] Confidential Containers. Welcome to confidential containers! https://confidentialcontainers.org/.

[52] gVisor. The container security platform. https://gvisor.dev/.

[53] Ethan G. Young, Pengfei Zhu, Tyler Caraza-Harter, Andrea C. Arpaci-Dusseau, and Remzi H. Arpaci-Dusseau. The true cost of containing: a gvisor case study. In *Proceedings of the 11th USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'19, page 16, USA, 2019. USENIX Association.

[54] Open Enclave. Open enclave sdk. https://openenclave.io/sdk/.

[55] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[56] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020.

[57] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

[58] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. Local model poisoning attacks to {Byzantine-Robust} federated learning. In *29th USENIX security symposium (USENIX Security 20)*, pages 1605–1622, 2020.

[59] Zhaoxian Wu, Qing Ling, Tianyi Chen, and Georgios B Giannakis. Federated variance-reduced stochastic gradient descent with robustness to byzantine attacks. *IEEE Transactions on Signal Processing*, 68:4583–4596, 2020.

[60] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. *Advances in Neural Information Processing Systems*, 33:16070–16084, 2020.

[61] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 691–706. IEEE, 2019.

[62] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the gan: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 603–618, 2017.

[63] Xinying Zou, Samir M Perlaza, Iñaki Esnaola, Eitan Altman, and H Vincent Poor. The worst-case data-generating probability measure in statistical learning. *IEEE Journal on Selected Areas in Information Theory*, 2024.

[64] Xinying Zou, Samir M Perlaza, Iñaki Esnaola, and Eitan Altman. Generalization analysis of machine learning algorithms via the worst-case data-generating probability measure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17271–17279, 2024.

[65] Olivier Beaumont, Thomas Lambert, Loris Marchal, and Bastien Thomas. Performance analysis and optimality results for data-locality aware tasks scheduling with replicated inputs. *Future Generation Computer Systems*, pages 582–598, 2020.

[66] Kamer Kaya, Bora Uçar, and Cevdet Aykanat. Heuristics for scheduling file-sharing tasks on heterogeneous systems with distributed repositories. *Journal of Parallel and Distributed Computing*, 67(3):271–285, 2007.

[67] Maxime Gonthier, Loris Marchal, and Samuel Thibault. Memory-aware scheduling of tasks sharing data on multiple gpus with dynamic runtime systems. In *2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 694–704, 2022.

[68] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[69] Ganesh Ananthanarayanan, Ali Ghodsi, Scott Shenker, and Ion Stoica. Effective straggler mitigation: Attack of the clones. In *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, pages 185–198, 2013.

[70] Orcun Yildiz, Shadi Ibrahim, Tran Anh Phuong, and Gabriel Antoniu. Chronos: Failure-aware scheduling in shared hadoop clusters. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 313–318. IEEE, 2015.

[71] Orcun Yildiz, Shadi Ibrahim, and Gabriel Antoniu. Enabling fast failure recovery in shared hadoop clusters: towards failure-aware scheduling. *Future Generation Computer Systems*, 74:208–219, 2017.

[72] Ifeanyi P Egwutuoha, David Levy, Bran Selic, and Shiping Chen. A survey of fault tolerance mechanisms and checkpoint/restart implementations for high performance computing systems. *The Journal of Supercomputing*, 65:1302–1326, 2013.

[73] Rachid Guerraoui, Nirupam Gupta, and Rafael Pinot. Byzantine machine learning: A primer. *ACM Comput. Surv.*, 56(7), April 2024.

[74] Rafael Stahl, Alexander Hoffman, Daniel Mueller-Gritschneder, Andreas Gerstlauer, and Ulf Schlichtmann. DeeperThings: Fully Distributed CNN Inference on Resource-Constrained Edge Devices. *International Journal of Parallel Programming*, 49(4):600–624, August 2021.

[75] Md. Maruf Hossain Shuvo, Syed Kamrul Islam, Jianlin Cheng, and Bashir I. Morshed. Efficient acceleration of deep learning inference on resource-constrained edge devices: A review. *Proceedings of the IEEE*, 111(1):42–91, 2023.

[76] Timothé Albouy, Davide Frey, Michel Raynal, and François Taïani. Asynchronous byzantine reliable broadcast with a message adversary. *Theoretical Computer Science*, 978:114110, 2023.

[77] Thomas Werthenbach and Johan Pouwelse. Towards sybil resilience in decentralized learning, 2023.

[78] Patient Ntumba, Nikolaos Georgantas, and Vassilis Christophides. Scheduling continuous operators for iot edge analytics with time constraints. In *2022 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 78–85, 2022.

[79] Patient Ntumba, Nikolaos Georgantas, and Vassilis Christophides. Adaptive scheduling of continuous operators for iot edge analytics. *Future Generation Computer Systems*, 158:277–293, 2024.

[80] Amaury Bouchra Pilet, Davide Frey, and François Taïani. Foiling sybils with haps in permissionless systems: An address-based peer sampling service. In *2020 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6, 2020.

[81] Can Karakus, Rahul Huilgol, Fei Wu, Anirudh Subramanian, Cade Daniel, Derya Çavdar, Teng Xu, Haohan Chen, Arash Rahnama, and Luis Quintela. Amazon sagemaker model parallelism: A general and flexible framework for large model training. *CoRR*, abs/2111.05972, 2021.

[82] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Towards understanding biased client selection in federated learning. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 10351–10375. PMLR, 28–30 Mar 2022.

[83] Mónica Ribero, Haris Vikalo, and Gustavo de Veciana. Federated learning under intermittent client availability and time-varying communication constraints. *IEEE Journal of Selected Topics in Signal Processing*, 17(1):98–111, 2023.

[84] Angelo Rodio, Francescomaria Faticanti, Othmane Marfoq, Giovanni Neglia, and Emilio Leonardi. Federated learning under heterogeneous and correlated client availability. *IEEE/ACM Transactions on Networking*, 32(2):1451–1460, 2024.

[85] Hubert Eichner, Tomer Koren, Brendan Mcmahan, Nathan Srebro, and Kunal Talwar. Semi-cyclic stochastic gradient descent. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1764–1773. PMLR, 09–15 Jun 2019.

[86] Yae Jee Cho, Pranay Sharma, Gauri Joshi, Zheng Xu, Satyen Kale, and Tong Zhang. On the convergence of federated averaging with cyclic client participation. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[87] Alfonso Pérez, Germán Moltó, Miguel Caballer, and Amanda Calatrava. Serverless computing for container-based architectures. *Future Generation Computer Systems*, 83:50–59, June 2018.

[88] Heiko Koziolek and Nafise Eskandani. Lightweight Kubernetes Distributions: A Performance Comparison of MicroK8s, k3s, k0s, and Microshift. In *Proceedings of the 2023 ACM/SPEC International Conference on Performance Engineering*, ICPE '23, pages 17–29, New York, NY, USA, April 2023. Association for Computing Machinery.

[89] Akash Network. Akash Network - Decentralized Compute Marketplace, July 2024.

[90] Mattia Fogli, Thomas Kudla, Bram Musters, Geert Pingen, Casper Van den Broek, Harrie Bastiaansen, Niranjan Suri, and Sean Webb. Performance Evaluation of Kubernetes Distributions (K8s, K3s, KubeEdge) in an Adaptive and Federated Cloud Infrastructure for Disadvantaged Tactical Networks. In *2021 International Conference on Military Communication and Information Systems (ICMCIS)*, pages 1–7, May 2021.