ODELIA

Open Consortium for Decentralized Medical Artificial Intelligence

HORIZON-HLTH-2021-CARE-05-02

Deliverable D1.1

REPORT ON SETTING UP SL CAPABLE COMPUTE INFRASTRUCTURE

Lead beneficiary	TUD
Author(s)	Oliver Saldanha
Reviewer	Alice Dudle (UZH)
Dissemination level	PU
Type	R
Delivery date	19 December 2024 (revision)

ODELIA is funded by the European Union's Horizon Europe Framework under Grant Agreement 101057091

Funded by the European Union

ODELIA

TABLE OF CONTENTS

Introduction 3
Hardware and Compute Resources 3
Software and System Configurations 3
VPN Setup 3
Data preprocessing 4
  Github repositories 5
Training 5
Lessons Learned and challenges faced from Setup with partner sites 6
  Technical Insights: 6
  Technical Challenges: 6
  Troubleshooting and Workarounds: 6
  Collaboration Challenges: 7
  Adaptations for Legacy Infrastructure: 7
Best Practices: 7
Initial training success 7
Conclusion 8

DISCLAIMER

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Health and Digital Executive Agency (HADEA). Neither the European Union nor the granting authority can be held responsible for them.

Introduction

This progress report outlines the project goals, infrastructure requirements, and key achievements in the ODELIA D1.1 phase, covering essential components such as partner site infrastructure setup, hardware and software configurations, and the implementation of Virtual Private Networks (VPNs). The VPN setup and adaptation were critical to enabling secure, decentralized communication across partner sites. Additionally, the report details the dataset used for initial training, the preliminary training processes undertaken, lessons learned, and a comprehensive conclusion on the deliverable's outcomes.

Hardware and Compute Resources

We recommend a minimum configuration of 8 CPU cores, 32 GB RAM, an NVIDIA GPU with 24 GB RAM, and 4 TB of storage running Ubuntu 20.04 LTS. For optimal performance, we suggest 16 CPU cores, 64 GB RAM, an NVIDIA GPU with 48 GB RAM, and 8 TB of storage. A high-speed network infrastructure, including routers, switches, and firewalls, is essential for secure, efficient communication. To ensure data redundancy and scalability, consider RAID configurations and modular storage solutions like NAS or SAN. For future-proofing, modular server racks and cloud-based solutions can be explored.

Software and System Configurations

We provided detailed specifications to consortium partners, including operating system requirements (Ubuntu 20.04 LTS), virtualization options (VMware, Hyper-V), and containerization with Docker for isolated environments. Essential software dependencies, such as PyTorch for machine learning, were also listed to ensure compatibility and smooth operation of the SL system.

VPN Setup

Firstly, the VPN setup as shown in (**Figure 1**) has been completed, which is a critical step in enabling secure and efficient communication between the different nodes. Good Access has been selected as the VPN provider due to its reliable Gateway with a static IP and support for Internet Key Exchange version 2 (IKEv2) and Open Source Virtual Private Network (OpenVPN) encryption.

On the one hand, a static IP address is important in VPNs because it provides a fixed, unique address that can be used to identify and connect to a specific device or network. In our setup, every center will have its own IP. This makes it easier to manage and configure the VPN, and ensures that we can maintain a consistent connection and access resources on the network without interruption. This VPN setup is ideal as it provides a mesh VPN architecture that aligns with the swarm architecture used. This ensures that traffic is not routed through a central host, which is important for security and efficiency.

On the other hand, encryption is important in VPNs because it provides a secure and private connection between the user's device and the VPN server, protecting their data from interception or unauthorized access. The detailed network diagram shown below has been prepared and submitted to each center's IT department for approval.

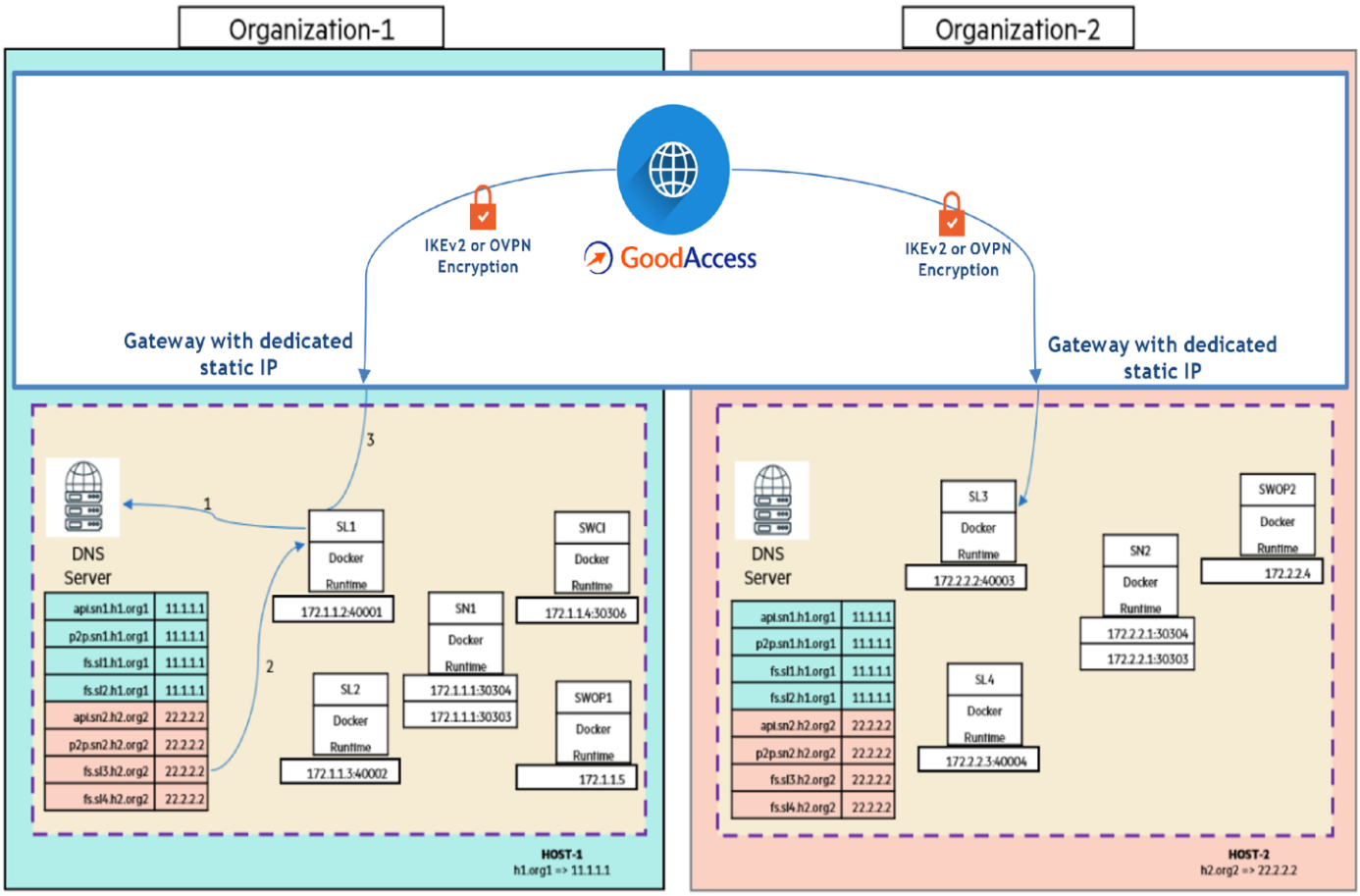

Figure 1: Visual representation of VPN setup in two centers with different IP addresses.

DATA PREPROCESSING

The DUKE dataset was used as a reference dataset to develop the pre-processing and training pipelines. The dataset is openly available and was collected between 2000 and 2014 at Duke University Hospital (Durham, North Carolina, USA), and includes 922 cases of biopsy-confirmed invasive breast cancer. The MRI protocol involved a T1-weighted fat-suppressed sequence (one pre-contrast and four post-contrast scans) and a non-fat-suppressed T1-weighted sequence.

Data pre-processing is essential to ensure accurate and reliable results in the training process. Data pre-processing as shown in (**Figure 2**) involves stratifying the data, cropping out the left and right breasts and saving them separately, as tumours mostly appear in one side of the breasts. In addition, the data was transformed from Digital Imaging and

Communications in Medicine (DICOM) files to Neuroimaging Informatics Technology Initiative (NIFTI) files. Indeed, DICOM files are not always compatible with all software tools and analysis methods, while NIFTI files are widely used in machine learning methods To standardize the data, all images are resampled to a uniform shape. The processed sequence is utilized to compute the pre-contrast and post-contrast subtraction images along with the resampled T1-weighted image.

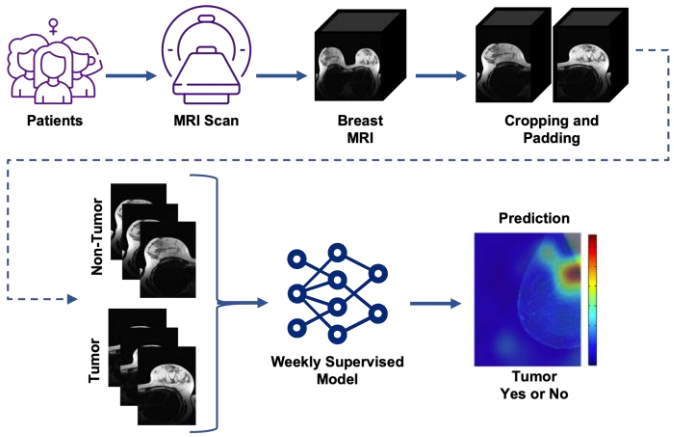

Figure 2: Schematic representation of the Deep Learning-based Weakly Supervised Learning (WSL) workflow for breast cancer tumour detection on Magnetic Resonance Imaging (MRI) data

Github repositories

A Github repository for the pre-processing pipeline has been created by UKA (https://github.com/mueller-franzes/odelia_breast_mri/tree/main/scripts/preprocessing). The training was conducted used the commercial swarm learning framework which used the Hewlett Packard Enterprise (HPE) implementation of Swarm Learning as the base, which is also publicly available at (https://github.com/HewlettPackard/swarm-learning/releases/tag/v2.2.0). A Github repository for model training and swarm learning integration of the pipeline has been created by UKA/TUD (https://github.com/KatherLab/swarm-learning-hpe).

TRAINING

This describes the training for Deep Learning-based Weakly Supervised Learning for breast cancer tumour detection on Magnetic Resonance Imaging (MRI) data. The pre-processed data from the DUKE dataset for MRI breast cancer is used to train a model for tumour detection. Multiple techniques such as 3-dimensional and 2-dimensional Convolutional Neural Network (CNN) and Attention-based Multiple Instance Learning (AttMIL) were tested. The Resnet-based 3D (CNN) model outperformed the other techniques and was thus chosen for the first experiments.

Lessons Learned and challenges faced from Setup with partner sites

The following section contains technical insights, collaborative challenges, and recommendations for future work.

Technical Insights:

- Hardware: The HPE platform's hardware requirements were stringent, leading to frequent troubleshooting. Considerations for hardware compatibility and high-performance specifications are essential for similar setups.
- Networking: The proprietary platform imposed network constraints that impacted latency and throughput. Future projects might benefit from open networking configurations to better support large datasets and complex models.
- Software: The need for licenses and proprietary dependencies limited scalability and raised challenges with automation. Open-source platforms offered greater flexibility and faster deployment.

Technical Challenges:

- Licensing Dependencies: Monthly license renewals added an administrative burden. This delayed operations, especially when coordinating with multiple partners.
- Network Latency: Frequent issues with VPN configurations due to complex requirements imposed by the proprietary software. This slowed communication with remote IT teams and impacted experiment runtime.
- Security Layers: Additional security checks for licensed software prolonged the setup phase and complicated on-site installations at partner institutions.

Troubleshooting and Workarounds:

- Software Patches: Regular software patches were required due to system instability, especially when handling updates and code synchronization across sites.
- Network Adjustments: Adjusted network settings to reduce latency, involving VPN and firewall reconfigurations.
- Hardware Adjustments: On-site visits became necessary to address hardware compatibility, which led to better alignment in hardware setups across partner sites.

Collaboration Challenges:

- Remote Coordination: Aligning schedules with remote IT teams often caused delays, especially when platform-specific expertise was needed for troubleshooting.
- Security Standards: Ensuring compliance with the proprietary system's security standards added extra coordination with each partner's IT department, creating bottlenecks in the setup process.
- Support Delays: Relying on HPE for support prolonged the resolution time, as detailed error logs were often inaccessible directly, complicating issue tracking and collaboration.

Adaptations for Legacy Infrastructure:

- Compatibility with Older Systems: Integrating swarm learning capabilities required hardware upgrades at some partner sites where legacy systems were still in use.
- Software Dependencies: Some partners with older software stacks faced challenges in meeting compatibility requirements for the proprietary platform, leading to additional troubleshooting efforts.

Best Practices:

- Streamlined Communication: Establish clear and consistent communication channels with external support teams to reduce delays in troubleshooting.
- Documented Setup Process: Creating a step-by-step guide for installation, addressing the proprietary platform's nuances, helped reduce on-site setup time.
- Proactive License Management: Preparing and distributing monthly licenses in advance to all partners minimized delays caused by last-minute renewals.
- Automated Quality Checks: Developing scripts for DICOM data preprocessing to avoid extensive back-and-forth, ensuring data consistency and reducing processing time.

Initial Training Success

Involving five nodes in joint training on the DUKE dataset, including VHIO, TUD, Radboud, and UKA, marks a major accomplishment in demonstrating effective collaboration and coordination across multiple centers. The project, split across five centers, including TUD, successfully completed swarm training for tumour detection, highlighting the robustness of the setup. The training was carried out using the commercial HPE-based swarm learning setup.

Setting up the infrastructure took an average of 2 hours per site, with an additional 20 minutes dedicated to ensuring that training data was accurately distributed and aligned across nodes. The swarm model took approximately 5 hours to converge, achieving performance on par with a centralized training setup where all data is pooled in one location.

Several complications arose during this process:

- Infrastructure Setup: Each site faced unique challenges in configuring the necessary infrastructure, including hardware compatibility and VPN setup, which required additional time and troubleshooting.
- Data Distribution: Ensuring consistent data alignment across nodes was critical, and any discrepancies led to delays and re-checks, adding complexity to the workflow.
- Network Latency and Throughput: The distributed nature of the setup introduced network latency issues, affecting data synchronization and increasing training time.

Finally, the ongoing process of other centers ordering hardware and awaiting delivery is a promising sign that the project will expand to include even more collaborators.

Conclusion

Overall, this progress report showcases the project's promising trajectory. By addressing the identified challenges and capitalizing on the successes, the ODELIA D1.1 is well-positioned to make significant contributions in the field of decentralized AI.