

D6.3 - AML-IP implementation - Revised M36

This deliverable explains in detail many aspects of the Algebraic Machine Learning - Integrating Platform (AML-IP) delving into its scope, purpose, design, and the current status of its implementation and deployment. The AML-IP is a multifaceted platform that must continuously adapt to support a dynamically evolving project. Consequently, the decisions, designs, and diagrams presented herein remain subject to modification in response to the project's evolving requirements.

The AML environment consists of nodes of different kinds that communicate with one another. These nodes compute different types of data which are published and shared with other nodes that request them in the AML ecosystem.

Several pivotal technologies underpin the development of the AML-IP:

- *Data Distribution Service (DDS)*: Given the distributed nature of the AML ecosystem, the use of DDS is proposed to enable real-time communication between AML nodes.
- *eProxima Fast DDS*: This C++ implementation of the DDS specification provides an Application Programming Interface (API) and a communication protocol to deploy a DDS environment.

The protocols developed on top of DDS that enable AML-IP node communications are:

- *DDS Topics*: AML-IP Topics within the DDS AML-IP network have a previous name mangling, resulting in DDS topics being named as `amlip::topic` within the DDS network.
- *MultiService Communication Protocol over DDS*: This innovative protocol capitalises on the advantages of DDS to create a specific protocol suitable for AML that responds to communication necessities specific to the ALMA project. This protocol targets the communication between several clients and servers in a way that only one server processes a request from any client, thus avoiding the use of a load balancer.
- *Remote Procedure Call*: Implemented based on DDS to meet the need to distribute requests and replies across an AML network effectively.

AML Available Resources

To support the AML project, several resources are available:

1. [ALMA Project Website](#)
2. [AML-IP Documentation](#)

3. [AML-IP GitHub Repository](#)

AML-IP Architecture

In order to divide the computational effort, to distribute the communication process and to create an scalable and widely usable architecture, the components of the AML-IP are divided in three categories:

1. *AML-IP Algebraic Node*: These nodes represent each of the instances of the AML.
2. *AML-IP Mediation Node*: These nodes facilitate seamless data transformation between different formats, enhancing interoperability within the AML ecosystem.
3. *AML-IP Agent Node*: Serving as the primary transport component, these nodes manage the dynamic discovery of AML-IP-Algs deployed within a LAN and the discovery of other AML clusters across WANs.
4. *AML-IP Discovery Node*: These components will be disseminated along the network and made accessible through the Internet. They manage the WAN discovery process along the whole AML environment, handling the discovery of independent AML clusters and efficiently connecting the AML-IP-Alg that must connect to each other.

One of the biggest strengths of AML is the great parallelization capabilities that its design enables, allowing different kinds of nodes to share their data in order to distribute the computational effort. These parallelization possibilities can be divided in two main types, depending on the scenario. Each of the scenarios correspond to a different use case and a parallelization method, but are not exclusive. All can coexist at the same time in the same network.

Collaborative learning scenario

Different AML-IP-Edge nodes collaborate on solving a common problem, each with the autonomy to work independently using distinct training datasets. This approach enables the creation of a more sophisticated and precise model by leveraging diverse data sources and perspectives. The output data being shared is independent of the dataset used to train the algorithm, so the training datasets can remain private while the resulting knowledge may be openly published and shared with the rest of the network. The AML-IP provides a collaborative and Internet wide network so remote and independent nodes have the possibility to work on the same problem and share their progress.

Workload distribution scenario

AML-IP-Edge nodes are responsible for training an AML model by handling individual data chunks. This training process is expensive and requires high computational power. However, it is also completely parallelizable. The AML-IP introduces a network architecture capable of efficiently distributing this computationally intensive workload across the network's nodes. This distribution occurs in a balanced and automated manner, irrespective of the nodes' physical location or ownership.

This eliminates the necessity for centralised load balancing mechanisms.

Monitor network state scenario

AML-IP-Edge nodes perform real-time monitoring, analysis, and debugging of an AML network. This scenario plays a critical role in ensuring the health and efficiency of the network by constantly knowing and analysing the status of the AML-IP nodes within it. Each AML-IP node is responsible for publishing its current status information, and this information is continually updated throughout the lifetime of each node.

Distributed inference scenario

AML-IP-Edge nodes primary objective is to efficiently distribute a substantial amount of data to remote nodes for the purpose of performing inference. This distribution aims to parallelize the inference process so that it does not interfere with other essential tasks that may need to run on the same devices. By implementing this measure, the system ensures the seamless execution of multiple tasks, optimising overall performance and resource utilisation.

Conclusions

This deliverable presented the design of the AML-IP. It is not intended to be the final design but a sketch on how it is going to be built, and it could change as the project advances depending on the necessities that may appear or the limitations encountered during the development process. However, it is intended to be a complete, robust and functional design fulfilling all the ALMA project requirements.