Cluster/ADAPT Interface Tool (CLAI)

Chalevin A., etamax space GmbH, European Southern Observatory, Garching, Germany.

The Cluster/ADAPT Interface (CLAI) is a Web application allowing the user to select ALMA observing blocks for manual and automatic pipeline reduction at one of the ALMA locations. CLAI produces prioritized lists of automatic and manual so-called Member Observing Unit Sets according to the locally configured filters and sends commands to the computing cluster to queue processing if such option is activated.

Introduction

The application was developed with a purpose to orchestrate pipeline processing of ALMA Member Observing Unit Sets. The algorithm was originally implemented in a Python tool and converted by us to Java. CLAI is a SpringBoot/Angular application, providing both GUI and REST interfaces to operate with pipeline executions.

There are several factors taken into account to select and prioritize ObsUnitSets. As a first approach, we pick ObsUnitSets with Pipeline or Manual processing recipes and a type of telescope array according to the local preferences. The obtained list of ObsUnitSets is then sorted by priority grade, observing cycle and time passed since the last observation was taken.

Sorted such way, ObsUnteSets then are being sent for automatic pipeline execution or provided as a list for manual processing.

1. Architecture

The application consists of the Java back-end server, written using Spring Boot framework and the Angular front-end component. The back-end server provides a public REST interface to let 3rd party scripts and applications manage pipeline executions [1] (see diagram 1). The Java server in its turn interacts with ALMA database server and, using ALMA message architecture [2] (ADAPT), with the XTSS (eXTernal State System) to set Member ObsUnitSets to the Processing state. The core functionality of the application is creating execution scripts to run Pipeline Driver (Python application) for every listed ObsUnitSet and pass them to distributed resource manager Torque to run pipeline executions.

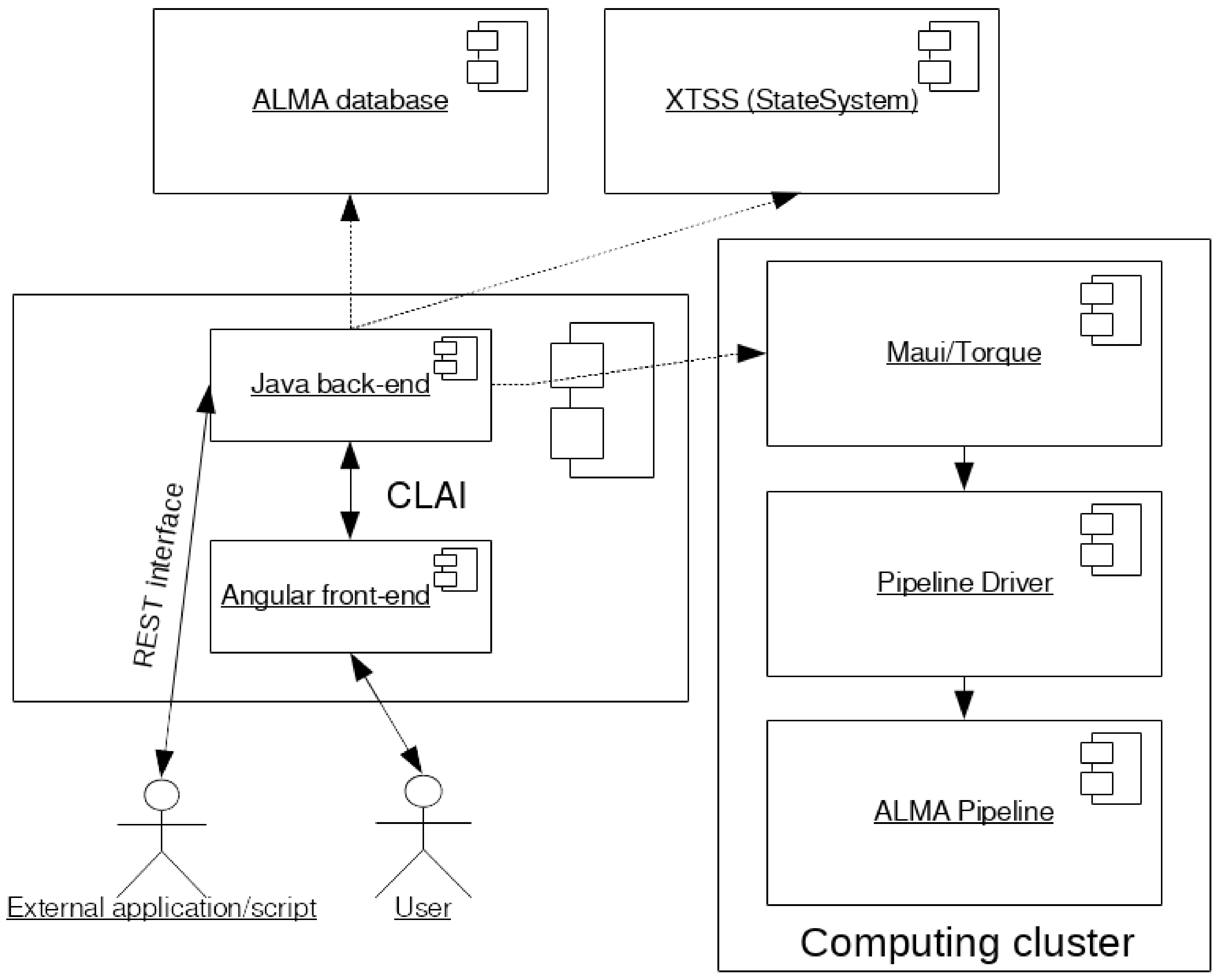


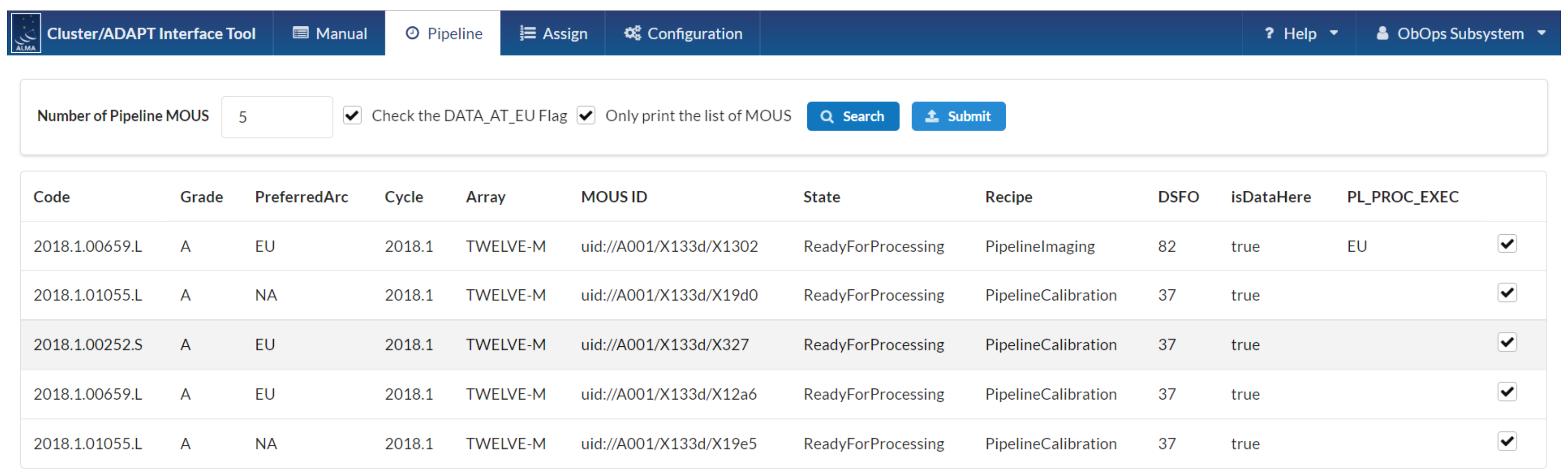
Fig.1. CLAI structure and interaction with other components.

2. Interface.

There are three endpoints available in the REST interface:

- “pipeline”, parameters "mode=try&pipe=5&checkData=true"
- “assign”, parameter ObsUnitSet ID
- “manual”, parameter a number of ObsUnitSets

They provide the functionality of pipeline batch (“pipeline”) and individual (“assign”) executions. Endpoint for manual processing (“manual”) provides just the list of ObsUnitSets which need to be processed.



Number of Pipeline MOUS 5 ☑ Check the DATA_AT_EU Flag ☑ Only print the list of MOUS

Code	Grade	PreferredArc	Cycle	Array	MOUS ID	State	Recipe	DSFO	isDataHere	PL_PROC_EXEC	
2018.1.00659.L	A	EU	2018.1	TWELVE-M	uid://A001/X133d/X1302	ReadyForProcessing	PipelineImaging	82	true	EU	☑
2018.1.01055.L	A	NA	2018.1	TWELVE-M	uid://A001/X133d/X19d0	ReadyForProcessing	PipelineCalibration	37	true		☑
2018.1.00252.S	A	EU	2018.1	TWELVE-M	uid://A001/X133d/X327	ReadyForProcessing	PipelineCalibration	37	true		☑
2018.1.00659.L	A	EU	2018.1	TWELVE-M	uid://A001/X133d/X12a6	ReadyForProcessing	PipelineCalibration	37	true		☑
2018.1.01055.L	A	NA	2018.1	TWELVE-M	uid://A001/X133d/X19e5	ReadyForProcessing	PipelineCalibration	37	true		☑

Fig.2. CLAI user interface.

The graphical user interface was written using Semantic UI framework and provides the same functionality as the REST interface with several interactive options, like a possibility to select which ObsUnitSets should be processed from the given list.

The future versions of the application will support additional configuration options for memory management and the possibility to run jobs in the serial or parallel modes.

References

1. P10.41 Joe Masters. Pipeline Calibration and Imaging for the ALMA Observatory

2. P9.1 A.M. Chavan. Re-engineering data processing for resilience: ALMA's new message-passing backbone.