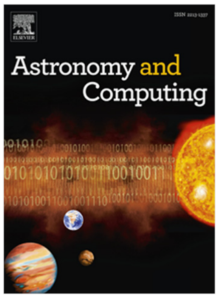
Accepted Manuscript



Corral framework: Trustworthy and fully functional data intensive parallel astronomical pipelines

J.B. Cabral, B. Sánchez, M. Beroiz, M. Domínguez, M. Lares, S. Gurovich, P. Granitto

PII: S2213-1337(17)30006-9
DOI: http://dx.doi.org/10.1016/j.ascom.2017.07.003
Reference: ASCOM 198

To appear in: *Astronomy and Computing*

Received date : 18 January 2017
Accepted date : 10 July 2017

Corral Framework: Trustworthy and Fully Functional Data Intensive Parallel Astronomical Pipelines

J. B. Cabral[a,b,*], B. Sánchez[a], M. Beroiz[c,d], M. Domínguez[a], M. Lares[a], S. Gurovich[a], P. Granitto[e]

[a] Instituto De Astronomía Teórica y Experimental - Observatorio Astronómico Córdoba (IATE–OAC–UNC–CONICET), Laprida 854, X5000BGR, Córdoba, Argentina.
[b] Facultad de Ciencias Exactas, Ingeniería y Agrimensura, UNR, Pellegrini 250 - S2000BTP, Rosario, Argentina.
[c] University of Texas Rio Grande Valley (UTRGV), One West University Blvd. Brownsville, Texas 78520, USA.
[d] University of Texas at San Antonio (UTSA), 1 UTSA Circle, San Antonio, TX 78249, USA.
[e] Centro Internacional Franco Argentino de Ciencias de la Información y de Sistemas (CIFASIS, CONICET–UNR), Ocampo y Esmeralda, S2000EZP, Rosario, Argentina.

Abstract

Data processing pipelines represent an important slice of the astronomical software library that include chains of processes that transform raw data into valuable information via data reduction and analysis. In this work we present Corral, a Python framework for astronomical pipeline generation. Corral features a Model-View-Controller design pattern on top of an SQL Relational Database capable of handling: custom data models; processing stages; and communication alerts, and also provides automatic quality and structural metrics based on unit testing. The Model-View-Controller provides concept separation between the user logic and the data models, delivering at the same time multi-processing and distributed computing capabilities. Corral represents an improvement over commonly found data processing pipelines in Astronomy since the design pattern eases the programmer from dealing with processing flow and parallelization issues, allowing them to focus on the specific algorithms needed for the successive data transformations and at the same time provides a broad measure of quality over the created pipeline. Corral and working examples of pipelines that use it are available to the community at `https://github.com/toros-astro`.

Keywords: Astroinformatics, Astronomical Pipeline, Software and its engineering: Multiprocessing; Design Patterns

1. Introduction

The development of modern ground–based and space–born telescopes, covering all observable windows in the electromagnetic spectrum, and an ever increasing variability interest via time–domain astronomy have raised the necessity for large databases of astronomical observations. The amount of data to be processed has been steadily increasing, imposing higher demands over: quality; storage needs and analysis tools. This phenomenon is a manifestation of the deep transformation that Astronomy is going through, along with the development of new technologies in the Big Data era. In this context, new automatic data analysis techniques have emerged as the preferred solution to the so-called “data tsunami” (Cavuoti, 2013).

The development of an information processing pipeline is a natural consequence of science projects involving the acquisition of data and its posterior analysis. Some examples of these data intensive projects include The Dark Energy Survey Data Management System (Mohr et al., 2008), designed to exploit a camera with 74 CCDs at the Blanco telescope to study the nature of cosmic acceleration; the Infrared Processing and Analysis Center (Masci et al., 2016), a near real-time transient-source discovery engine for the intermediate Palomar Transient Factory (iPTF Kulkarni, 2013); and the Pan-STARRS PS1 Image Processing Pipeline (Magnier et al., 2006), performing the image processing and analysis for the Pan-STARRS PS1 prototype telescope data and making the results available to other systems within Pan-STARRS and Vista survey pipeline that includes VIRCAM, a 16 CCD nearIR camera for the VISTA Data flow system Emerson et al. (2004) . In fact, the implementation of pipelines in Astronomy is a common task to the construction of surveys (e.g. Marx and Reyes, 2015; Hughes et al., 2016; Hadjiyska et al., 2013), and it is even used to operate telescopes remotely, as described in Kubánek et al. (2010). Standard tools for pipeline generation have already been developed and can be found in the literature. Some examples are Luigi[1], which implements a method for the creation of distributive pipelines; OPUS (Rose et al., 1995), conceived by the *Space Telescope Science Institute*; and more recently Kira (Zhang et al., 2016), a distributed tool focused on astronomical image analysis. In the experimental sciences, collecting, pre-processing and storing data are common recurring patterns regardless of the science field or the nature of the experiment. This means that pipelines are in some sense re-written repeatedly. A more efficient approach would exploit existing resources to build new tools and perform new tasks, taking advantage of established procedures

*Corresponding author
Email address: `jbcabral@oac.unc.edu.ar` (J. B. Cabral)

[1] Luigi: `https://luigi.readthedocs.io/`