Running Roman pipelines in NASA cloud with Apache Airflow and Kubernetes

Emmanuel Joliet

[1] *IPAC/Caltech, Pasadena, CA, USA;* ejoliet@caltech.edu

Abstract. The Nancy Grace Roman Space Telescope is a NASA observatory to unravel dark energy and dark matter, search for and image exoplanets, and explore infrared astrophysics. This talk highlights the solution adopted by the Roman SSC team at Caltech/IPAC to run data pipelines using Airflow and Kubernetes in the NASA cloud.

1. The Roman mission

The Nancy Grace Roman Space Telescope is a NASA observatory currently in development. It is designed to answer fundamental questions in the areas of dark energy, exoplanets, and astrophysics. Roman will have two instruments, the Wide Field Instrument (WFI) and the Coronagraph Instrument. As the primary instrument, the WFI will observe the light from a billion galaxies over the mission and perform a microlensing survey of the inner Milky Way to find 2,600 exoplanets. The Coronagraph Instrument will perform a technological demonstration preforming high contrast imaging of individual nearby exoplanets. IPAC is supporting the science operations together with JPL, STScI and GSFC. Among several other items, the primary responsibilities for the Roman Science Support Center (SSC)[1] at IPAC includes science data processing for the WFI microlensing survey and all spectroscopic observations in the NASA cloud.

1.1. Key facts

Key facts of the mission are:

- Launch to L2 no later than May 2027
- Lifetime: 5 years
- Primary mirror is 2.4 meters (7.9 feet) in diameter
- Instruments
  - Wide-Field Instrument (0.48um to 2.3um)
  - Coronagraph Instrument (0.55um to 0.86um)
- Roman SSC at IPAC to produce high level sci data products

[1] https://roman.ipac.caltech.edu

1.2. Roman SSC responsibilities

Roman SSC science data processing pipelines in the cloud are:

- WFI Microlensing Science Operations System (MSOS)
- WFI Grism and Prism Data Processing System (GDPS)

Roman SSC will produce high level science data products and support Galactic Bulge Time Domain Survey and High Latitude Wide Area, High Latitude Time Domain and Guest Observer surveys.

- MSOS: Microlensing survey processing pipeline producing daily light curves, seasonal objects and events catalogs among other products. 200 millions object catalogs, Microlensing events catalog, Light Curves
- GDPS: Produces decontaminated 2D and 1D extracted spectra and spectral redshifts among other products.

Other SSC main systems: Coronagraph Instrument Operations and Data Management System, Roman Telescope Proposal System, and Community Engagement. Data volumes:

- MSOS: 5 TB/daily 2PB over mission lifetime (6 seasons of 62 days each)
- GDPS: 1.5TB/daily during survey (unsure since TBD observations)
- CGI - 1TB over tech demo

Below are examples of simulated spectra and microlensing event that would be the science products coming out of the pipelines from MSOS (see figure 1) and from GDPS (see figure 2).

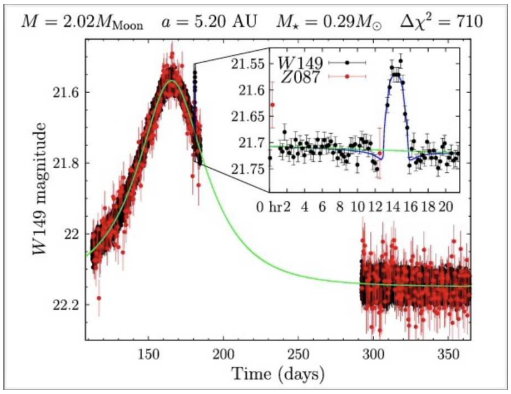

Figure 1. Simulated Roman microlensing event (2 seasons) with a planetary anomaly (Penny et al. (2019))

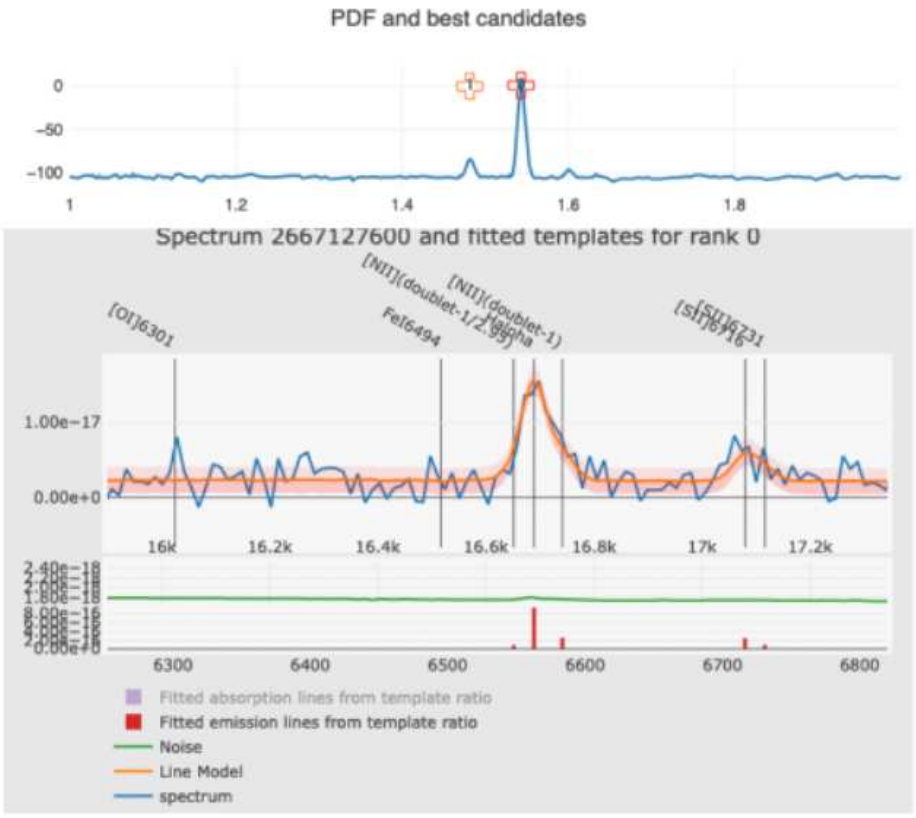

Figure 2. Example of redshift PDF visualization (top) and spectral fitting (bottom) in the 1D pipeline. In the bottom panel, the data are shown in blue and fit to the continuum and the spectral features (along with the uncertainty) are shown in red. The fit to the emission lines are used to generate the zPDF shown in the top plot, here with a strong peak in the solution at a redshift z 1.55. A weaker, secondary maximum is also seen at z 1.48.

2. NASA Cloud platform

The NASA Mission Cloud Platform (NMCP) uses AWS[2] services to run mission work including running the science pipelines. AMIs, VPCs, ACL and few other cloud related environment is managed by NMCP. Roman SSC sets the roles, services and manages deployments. The Docker images are stored and managed from AWS ECR service. The containers are provisioned and orchestrated by Kubernetes cluster using the Elastic Kubernetes Service (AWS EKS). The Roman science products files are stored and indexed in S3 buckets. Databases Aurora RDS PostgreSQL are used to host the Airflow metadata[3] and pipeline processing needs.

[2]https://aws.amazon.com

[3]https://docs.astronomer.io/learn/airflow-components#core-components

3. Software engineering

Roman SSC uses a GitOps approach for defining and deploying the infrastructure and resources. We apply DevOps best practices to a CI/CD chain, with version control and infrastructure-as-code automation. Our Python ecosystem includes asdf, astropy, scipy, pydantic, dask, boto3, matplotlib, pytest/tox, and Airflow API, among others. The C/C++ codebase makes use of numerical and test unit libraries. The Code, DAGs and AWS scripts to support running the pipelines are under Configuration Management in IPAC private GitHub repositories. Docker images are used as a release artifacts to be distributed in EKS cluster and run as PODs. Security issues and common vulnerabilties exposure (CVEs) are scanned on GitHub code and on containers hosted in AWS ECR with Snyk tool. A Jenkins server on an EC2 instance is running builds, tests on code changes and releases. The Apache Airflow frontend UI allows operators to monitor and troubleshoot the data pipelines, providing insights into Directed Acyclic Graphs (DAGs) and DAG runs.

4. Apache Airflow in the cloud

The Roman SSC science data processing pipeline tasks are scheduled and run in a Kubernetes cluster (AWS EKS) using the Apache Airflow[4] scheduler and UI dashboard. The task workflows are defined as Direct Acyclic Graphs (DAGs) using the Airflow task API. The infrastructure-as-code scripts (CloudFormation stacks[5] and Helm charts are used to manage software and services deployments into Kubernetes such as Roman software containers, Apache Airflow, Grafana and Prometheus. Docker images registry (AWS ECR) helps storing, distributing and versioned the science software under GitHub. The Airflow *KubernetesPodOperator*[6] operator is used to define the task and its resources. Node group consists of EC2s 'aml2', with ability to scale vertically and horizontally

5. Conclusion

Roman SSC at IPAC/Caltech uses Apache Airflow scheduler and frontend UI to define, deploy, run and monitor pipeline tasks on a Kubernetes cluster in the NASA cloud, making use of several other AWS services for storage, data transfer and scalability purposes.

Acknowledgments. I would like to thank Patrick Lowrance and Keto Zhang from Roman SSC at IPAC/Caltech who reviewed the poster/paper.

References

Penny, M. T., Gaudi, B. S., Kerins, E., Rattenbury, N. J., Mao, S., Robin, A. C., & Calchi Novati, S. 2019, ApJS, 241, 3. 1808.02490

[4]https://airflow.apache.org

[5]https://aws.amazon.com/cloudformation/

[6]https://airflow.apache.org/docs/apache-airflow-providers-cncf-kubernetes/stable/operators.html