Empowering Researchers with Open-Source Tools: A JupyterHub and Docker-Integrated Planetary Data Processing Environment.

Giacomo Nodjoumi[1], Carlos Brandt[1], Javier Suárez-Valencia[1], Erica Luzzi[2], Riccardo Pozzobon[3], Angelo Pio Rossi[1]

[1]Constructor University Bremen, Campus Ring 1, Bremen, Germany (gnodjoumi@constructor.university),

[2]University of Southern Mississippi, Hydrographic Science Research Center, USA,

[3]Dipartimento di Geoscienze, Università di Padova, Italy.

Abstract

Planetary data, acquired from various space missions, provide valuable insights into the composition, structure, and evolution of our Solar System. However, effective analysis and interpretation of this vast trove of data require specialized tools and expertise. Here, we present an innovative approach for deploying and managing planetary data processing environments based on ISIS, ASP, and Python, empowering researchers with a user-friendly, scalable, and customizable solution to conduct high-level research with planetary data.

Introduction

Planetary data are typically available in a range of formats, from raw sensor measurements to processed and calibrated products. Access to such data is facilitated by planetary data archives, e.g., the NASA Planetary Data System (PDS) [1], the European Space Agency (ESA) Planetary Science Archive (PSA) [2], Mars System of Information (MarSI) [3], and Multi-purpose Advanced Tool for Instruments of the Solar System Exploration (MATISSE) [4]. However, transforming the data into meaningful scientific results often requires the use of specialized software packages and a certain level of technical proficiency.

Multiple efforts are currently being made to provide access to Analysis Ready Data (ARD) [5]–[7] and interactive web-based analysis tools [8]. However, most of the high-end products are still processed manually through widely used open-source tools such as a) the Integrated Software for Imagers and Spectrometers (ISIS) [9-11], developed by the U.S. Geological Survey (USGS) Astrogeology Research Program, which is the reference open-source software suite for processing and analyzing planetary image and spectral data, and b). the NASA Ames Stereo Pipeline (ASP) [12], which is the reference open-source tool specifically designed for generating high-resolution digital terrain models (DTMs) from stereo imagery. Nevertheless, these powerful tools require installation, configuration, and familiarity with UNIX-based systems, which can pose a barrier for various researchers.

Proposed Solution

To address these challenges and simplify access to these powerful tools, we developed a user-friendly and scalable solution for deploying and managing planetary data processing environments based on ISIS, ASP, and Python. Our approach leverages Docker containers [13], JupyterHub [14], and a dedicated yet highly customizable docker image, named ISIS-ASP-GISPY, which includes a collection of commonly used Python packages (GISPY) that users can further implement to accommodate their needs on top of ISIS and ASP installation.

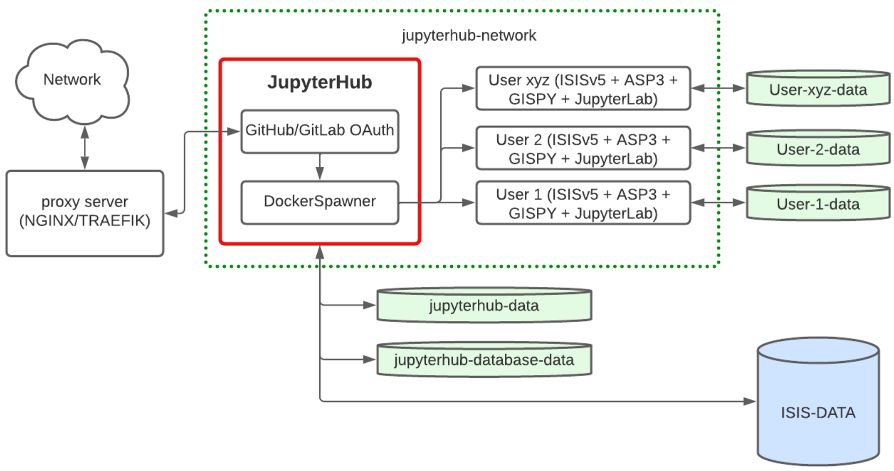

Figure 1. Simplified schematics of the GMAP JupyterHub architecture.

Docker containers provide a lightweight and isolated environment for running software applications without dependencies and conflicts with the host system. JupyterHub is a web-based interactive computing platform enabling users to launch and manage Jupyter notebooks, which are interactive documents able to combine code, output, and narrative text. GISPY (Geospatial Python) is a collection of Python packages commonly used for geospatial data analysis. Figure 1 shows the schematics of the proposed system.

Deployment and Usage

The deployment of our solution is straightforward and requires minimal configuration. A single script automates the creation of Docker volumes, networks, and images, simplifying the setup for system administrators. Additional parameters are necessary for advanced configurations such as the web-service publication, the user access restriction, and resources limitation. Users can access the JupyterHub instance through a web interface, where they can create and manage their Jupyter notebooks.

Advantages and Impact

Our proposed approach offers several advantages over traditional deployment methods:

- Simplified Deployment and Maintenance: The Dockerized environment eliminates the need for complex installations and configurations, reducing overhead for system administrators.
- User-Friendly Interface: The Jupyter Hub interface provides a familiar and intuitive platform for users to interact with the data processing tools, catering to researchers with varying levels of technical expertise.
- Scalability: The system can be easily scaled to accommodate a growing number of users, making it suitable for collaborative workgroups, classrooms, and workshops.
- Customization: The Docker images and Python packages can be customized to meet specific user requirements, enabling the incorporation of additional tools and functionalities.

The implementation of our solution has been successfully tested and the release version is available on Zenodo and GitHub. Ongoing development efforts are focused on enabling users to reconnect and recover the state of their notebooks after a disconnection, which is a crucial feature for long-running processing tasks.

The service has also been deployed as a web service at https://jupyter.europlanet-gmap.eu/: https://jupyter.europlanet-gmap.eu/. With prior registration, users can access the service and process data. The service is part of the Europlanet GMAP (Geological Mapping of Planetary Bodies) project, which aims to develop and implement planetary geological mapping techniques. For more information about GMAP, please visit: https://europlanet-gmap.eu/

Conclusion:

The proposed planetary data processing environment offers a user-friendly, scalable, and customizable solution for analyzing planetary data using ISIS, ASP, and Python. By lowering the barriers to entry and providing a streamlined workflow, this approach empowers researchers with a powerful toolkit for extracting knowledge from planetary data, furthering our understanding of the cosmos.

References:

[1] PDS Geosciences Nodes, ‘PDS Geosciences Node Orbital Data Explorer (ODE)’. 2020. Accessed: Nov. 10, 2020. [Online]. Available: https://ode.rsl.wustl.edu/

[2] European Space Agency (ESA), ‘ESA Planetary Science Archive (PSA)’, 2022. https://archives.esac.esa.int/psa/ (accessed Mar. 21, 2022).

[3] C. Quantin-Nataf et al., ‘MarsSI: Martian surface data processing information system’, Planetary and Space Science, vol. 150, pp. 157–170, Jan. 2018, doi: 10.1016/j.pss.2017.09.014.

[4] A. Zinzi et al., ‘MATISSE 2.0: The SSDC Webtool for Integrated Planetary Science Analysis’, LPI Contributions, vol. 2549, p. 7018, 2021.

[5] R. L. Fergason, M. A. Hunter, J. R. Laura, and T. M. Hare, ‘Analysis Ready Data Available Through the SpatioTemporal Asset Catalog (STAC) Specification: Investigating the Application to Planetary Data’.

[6] T. M. Hare, B. J. Thomson, L. R. Gaddis, J. Stopar, B. A. Archinal, and J. R. Laura, ‘Building a lunar spatial data infrastructure (SDI)’, presented at the 5th Planetary Data and PSIDA, 2021.

[7] J. R. Laura and R. A. Beyer, ‘Knowledge Inventory of Foundational Data Products in Planetary Science’, Planet. Sci. J., vol. 2, no. 1, p. 18, Jan. 2021, doi: 10.3847/PSJ/abcb94.

[8] C. Fraiser and L. Keszthelyi, ‘Planetary Image Editor I/O-Web Service’, presented at the 5th Planetary Data and PSIDA, 2021.

[9] L. Gaddis et al., ‘An Overview of the Integrated Software for Imaging Spectrometers (ISIS)’, in Lunar and Planetary Science Conference, Mar. 1997, p. 387.

[10] J. A. Anderson, S. C. Sides, D. L. Soltesz, T. L. Sucharski, and K. J. Becker, ‘Modernization of the Integrated Software for Imagers and Spectrometers’, in Lunar and Planetary Science Conference, Mar. 2004, p. 2039.

[11] J. Laura et al., Integrated Software for Imagers and Spectrometers. Zenodo, 2022. doi: 10.5281/zenodo.6329951.

[12] R. Beyer, O. Alexandrov, and S. McMichael, NeoGeographyToolkit/StereoPipeline: NASA Ames Stereo Pipeline 2.6.1. Zenodo, 2018. doi: 10.5281/zenodo.1345235.

[13] D. Merkel, ‘Docker: lightweight Linux containers for consistent development and deployment’, Linux journal, vol. 2014, no. 239, p. 2, 2014.

[14] T. Kluyver et al., ‘Jupyter Notebooks – a publishing format for reproducible computational workflows’, in Positioning and Power in Academic Publishing: Players, Agents and Agendas, 2016, pp. 87–90.