Unified Planetary Coordinates Database Refactor.

A. R. Sanders[1], S. Akins[1], D. P. Mayer[1], E. A. Bovre[1], J. Laura[1], L. Gaddis[1]. U. S. Geological Survey, Astrogeology Science Center 2255 N. Gemini Dr., Flagstaff, AZ (arsanders@usgs.gov)

Introduction: The Unified Planetary Coordinates (UPC) database provides a searchable interface of the labels, camera statistics, and vector-based footprint geometry for image files archived by the Planetary Data System (PDS) Imaging Node at the USGS. The original development of the UPC database was motivated by a need from the scientific community for a way to access planetary image data collected by different sensors that has been archived in disparate coordinate systems [1]. Users can query the UPC database through the USGS PILOT [2] interface (https://pilot.wr.usgs.gov/) and choose to either download raw products directly or request that the data be processed in ISIS into map-projected products via the Projection on the Web (POW) web service. The collection of software used to construct and maintain the UPC database, and to facilitate image processing through the POW [3] and Map-a-Planet 2 (MAP2) web services is called PDS-Pipelines.

In its current form, the UPC database tracks all keywords from the original product labels as well as keywords about product viewing and lighting geometry derived from SPICE data. Although this level of detail permits very generalized queries, the volume of metadata results in slow queries and requires each mission instrument to be uniquely managed (as the associated metadata are unique to a particular sensor). The inefficiency and complexity of the existing architecture have motivated a refactor that comprises the following tasks:

- Translation from Perl to Python
- Automation of end-to-end processing
- Optimization of the database
- Containerization

This refactor serves to both directly and indirectly benefit the scientific community by improving the efficiency and scalability of the UPC database, which plays a critical role in the support of utilities such as POW, MAP2, and PILOT. Additionally, the refactor will greatly improve the extensibility, portability, and maintainability of the PDS-Pipelines toolchain that facilitates image processing capabilities.

Translation: This portion of the refactor involves the translation of the existing, Perl-based codebase to a Python implementation. This translation is necessary to increase the extensibility of the PDS pipeline as well as allowing access to more modern libraries and frameworks that can be used to decrease both the conceptual and computational complexity of our work.

Additionally, as part of the shift from Perl to Python, we have taken the opportunity to modernize the software engineering practices used in the project. One notable change is the shift from SVN to Git (https://github.com/USGS-Astrogeology/PDS-Pipelines), which not only better matches the needs of our organization, but serves to increase the visibility of our project within scientific and open source communities.

Automation: The initial implementation of the PDS pipelines software required a user to manually add files to the UPC database and track files that were due for data integrity checks. However, due to the increasing volume of data, the manual process has become intractable, and it is necessary to develop an automated means of performing data integrity and UPC processing.

In order to address the shortcomings of the existing system, we implemented the new system such that both data integrity and UPC ingestion steps can be fully automated. In the case of data integrity, the system checks the most recent processing date for each archive, compares those dates to a policy, and reprocesses volumes as necessary. Similarly, the system autonomously identifies, ingests, and processes any files that are eligible for UPC processing. In this way, the new system effectively removes the need for a human in the loop for both UPC processing and data integrity management.

Optimization: One of the foremost issues of the existing PDS Pipelines system is the inefficiency of the database. For the existing implementation, mission scientists determined that all label and camera keywords should be made available for searching within the database. To accommodate mission-specific keywords, values are spread

throughout multiple tables and required manually specified configurations and mission-specific management systems. Consequently, database queries involve the combination of values from multiple tables which results in costly and inefficient database searches.

In practice, only a subset of keywords that are common across all datasets is readily exposed in PILOT. This subset includes all keywords necessary for identifying image products geographically, in time, and with particular viewing and lighting geometries. This means that the current practice of tracking additional, instrument-specific keywords in the UPC database is causing queries to be slow without adding value for users.

We believe that the keywords currently exposed in PILOT are sufficient for most users to identify products that meet their needs. In order to support users who desire a more granular search capability, we will be adding a table to the database that stores all available keywords in a lightweight, human-readable data format known as JSON. This format allows for the specification of key/value pairs and can be easily parsed and queried, which makes it particularly well-suited for storing keyword information.

This will have the practical benefit of speeding up queries in PILOT using the currently-exposed, common keywords, while allowing users to download a JSON representation of all available keywords from products in their search results, and perform advanced queries using their own tools.

Containerization: The process of containerization entails the packaging of code, dependencies, and configurations into lightweight, distributable units of software. These units, which are called "images" are instantiated into running containers through the use of container management engines such as Docker, Apache Mesos, or LXC. These engines are responsible for providing an abstraction layer between instantiated containers and a host machine's operating system. In this way, a running container is isolated from the host machine's computational environment (operating system, existing packages, etc.) such that the software is guaranteed to perform in a uniform manner despite potential differences in host machines.

This portion of the refactor will involve the specification of a Docker image that includes the PDS-Pipelines software along with a PostgreSQL database instance. Not only does this step contribute to the modernization and portability of the product, but it also allows for the creation of ad-hoc, configurable instances of the database that can be tailored to the specific needs of an organization or end user.

Future Direction: The refactored UPC database will enable improved access to PDS Imaging Node holdings at the USGS through existing web services and serve as an example implementation for external users who wish to stand up their own planetary image database. At the conference we will provide a status report on the UPC refactor, including technical examples of the revised database architecture, and describe how users can deploy their own containerized instance of a UPC database.

Acknowledgments: This project is supported by NASA's Planetary Data Science (PDS) program and by the researchers on the teams that it comprises. More information can be found at https://pds.nasa.gov. Additionally, the authors wish to acknowledge Bob Sucharski for his guidance, development efforts, and contributions to the early stages of the PDS Pipelines and Database project.

References:

[1] Akins, S. W. et al, Status of the PDS Unified Planetary Coordinates Database and the Planetary Image Locator Tool (PILOT) (2009), LPSC XL, Abstract #2002. [2] Bailen, M. S. et al, Using the PDS Planetary Image Locator Tool (PILOT) to Identify and Download Spacecraft Data for Research (2013), LPSC XLIV, Abstract #2246. [3] Hare, T.M. et al, Map Projection Web Service for PDS Images (2013), LPSC XLIV, Abstract #2068.