Heliophysics API (HAPI) and Related Software

Bob Weigel, Jon Vandegriff, Jeremy Faden, Aaron Roberts, Todd King, Nand Lal, Bobby Candey, Bernie Harris, Doug Lindholm, Larry Brown

Overview

- HAPI is a specification for serving time series data
- Main entry point for overview https://hapi-server.org/
- Main entry point for specification and software https://github.com/hapi-server/
- Specification, clients, and servers are mature

Motivation

- Analysis of data from APIs from N data providers should not require learning how to use N APIs and N response formats.
- As a user, my data preparation end goal is an array of data in my analysis software.
- With HAPI, a user can execute the following in MATLAB, IDL, Python, Javascript, and Java

```
server     = SERVER
dataset    = DATASET
start      = START
stop       = STOP
parameters = PARAMETERS
data, meta = hapi(server, dataset, parameters, start, stop)
```

For `SERVER`, choose from 8 possibilities: AMDA, CCMC/ISWA, CDAWeb, INTERMAGNET, LISIRD, SSCWeb, Ulowa/Das2, ViRES for Swarm, (several more in development)

`data` and `meta` are data structures appropriate to given language

Session-related Aspects

Discuss tooling and user-facing aspects of analysis, i.e., things that scientists will use.

Existing software packages and single-user codebases typically have three components

A. Reading data from a limited set of data providers
B. Analysis
C. Visualization

The HAPI project priority is A.

Session-related Questions

***Goal**: Full use of data from great observatory platforms will require advanced analysis capabilities. Many existing analysis tools were created before the era of large constellations and big data.*

What analysis capabilities exist now, and how can these be evolved or transformed to meet the challenges that we see (bigger data, open science requirements, the need to compare data from different missions / science areas) and also the new challenges that will emerge?

Requires development of

1. Software libraries
2. Standards
3. Services

that accounts for changings in the science community and changes in funding availability. To understand how complex this is, consider the question of identifying what capabilities exist now. PyHC has a curated list of Python-specific tools. But there is much more out there in Python and other languages.

Session-related Questions

How do we make software and data analysis an enterprise-level activity instead of being a collection of mission-specific tools?

1. Incentive. Community collaboration is more efficient in long term but can be less efficient in short term.
2. Training. Developing tools for re-use. Managing a software project and community contributions. It is easy to make software “open” (just post it). The rest is challenging.
3. Lower barriers to collaboration.
4. Research - What has worked in the past? Why were mission (and, often, instrument-specific tools) developed?

Session-related Questions

How do we make software and data analysis an enterprise-level activity instead of being a collection of mission-specific tools?

4. Research - What has worked in the past? Why were mission (and, often, instrument-specific tools) developed?

Case study from HAPI development:

- Support for work through NASA grants
- Scientists and software engineers who all have worked on related development or used related APIs. Encouraged community participation
- Developed standards (this takes a long time!)
- Result was buy-in and use by external data providers