HAPI Cache: Software to Aid Caching of Pertinent Science Data

Norberto Lopez[1], Jeremy Faden[2], Robert Weigel[3], Jon Vandegriff[1], Alex Antunes[1]
1. Johns Hopkins Applied Physics Laboratory, 2. Cottage Systems, 3. George Mason University

Session: IN31C: Coping with the Data Deluge Poster
Poster Number: IN31C-2031

Time: 2024 Dec 11 (08:30 – 12:00)
Location: Convention Center, Hall B-C (Poster Hall)

Abstract

The Heliophysics Application Programming Interface (HAPI) provides a specification for accessing time-based data in a generic fashion. Using data sources that satisfy the HAPI specification provides for natural mechanism to access a subset of the data either chronologically or by reduction of necessary parameters.

The HAPI-Cache software enables caching of data retrieved from a HAPI source. Cached data offers the usual benefits of improved performance and offline access. HAPI-Cache provides a mechanism to control the size, parameter selection, and freshness of science data. An additional benefit is a standardized read and write capability to the cache, so multiple HAPI client implementations (for example, a Java application and Python analysis) can use the same cache.

More about HAPI:
https://hapi-server.org

Check out the HAPI Cache on github:
https://github.com/hapi-server/cache-tools

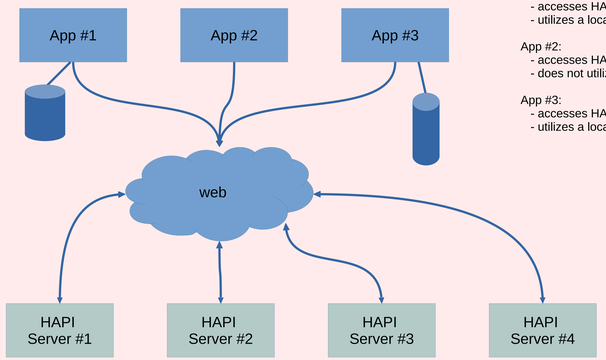
A Definition of:

Data Deluge

A scenario where more data is generated than can be successfully and efficiently managed or capped.

Source:
https://www.hpe.com/us/en/what-is/data-deluge.html

Without HAPI Cache



Example Scenario:

App #1:
- accesses HAPI servers #1, #2, #4
- utilizes a local custom cache

App #2:
- accesses HAPI servers #2 and #3
- does not utilize a local custom cache

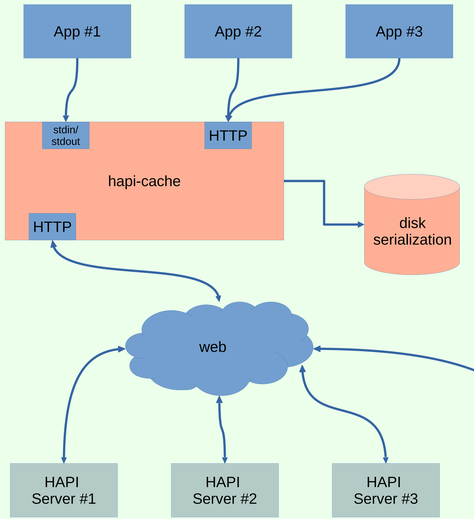
App #3:
- accesses HAPI servers #1, #3, #4
- utilizes a local custom cache

Potential Issues:

Apps #1 and #3:
- Local storage for each app may have a lot of duplicate cached data
- Local storage cached data for the same data sets may not be equivalent, depending on when the cached data was refreshed or fetched

App #2
- May not work (or stop working) if:
  → The network goes down
  → Server #2 or server #3 is inaccessible
- May spend a lot of time and bandwidth (re)fetching data

With HAPI Cache



Example Scenario:

Apps #1, #2, #3:
- access a single instance of a common shared HAPI Cache
- App #1 utilizes stdin/stdout protocol
- Apps #2 and #3 utilize HTTP protocol

Improvements:

Apps #1, #2, and #3:
- No duplicate cached data
- Single source of truth (cached data) means each app has the same equivalent cached data
- Reduction of network bandwidth
- Reduction of disk space requirements
- Automatic caching capability

Benefits For Scientists

- Allows offline access
- Faster data access time
- Reduces disk space needs (multiple programs use the same cache)
- Ensures the various app's caches are synchronized
- Reduces network bandwidth usage
- Reduces dependency on 3rd party HAPI servers

Provides fine grained controls on when the data is refreshed

Benefits For Developers

Provides a caching mechanism without the need to write from scratch

No need to design a serialization cache capability

Continual improvements from the HAPI community

Reduces development time

Ultimately, the HAPI cache allows one to focus on the problem domain instead of baseline infrastructure development.

About that data deluge...

- Where is my cached data?
- What is stored in my cache?
- What format is the cached data?
- Are the various app's caches synchronized?
- How old is the data in the cache?
- How is the cache cleared, trimmed, or refreshed?
- How do I reuse this cached data for other apps?

Consider the number of questions from the sampling to the left that could be asked of a single application…

As more applications are utilized, the effort to answer these questions and manage each individual application's cache is effectively a multiplying problem.

HAPI Cache provides a standardized method to aid with the management of these cached data stores.

Currently in progress…

Implementation of specifics for the HTTP protocol

Engineering of a mechanism to allow for:
→ fine grained cache controls
→ clearing the cache
→ refreshing the cache
→ etc

Are you ready to try out HAPI Cache?

Check out the binary build:
https://cottagesystems.com/~jbf/hapi/p/hapi-cache

- or -

Contact faden@cottagesystems.com to enable in Autoplot
https://autoplot.org