Feedback on TAP Implementation at NASA Astronomical Virtual Observatory (NAVO) Archives

Bruce Berriman, Thomas McGlynn
On behalf of HEASARC, IRSA/IPAC and MAST.

Summary

- NASA committed to providing TAP interfaces.
- TAP services implemented at HEASARC and IRSA
  - MAST service under development; delivery later in 2015
  - CXC release including Chandra source catalog.
- Concerns over complexity of ADQL
- Considerable development costs, yet usage of HEASARC service (released in 2012) is low.
  - How to increase usage? New clients needed?

HEASARC TAP Implementation

- Released 2011-2012
- One TAP service including all 860 HEASARC tables
- Single registry entry describing only the basic TAP URL. Tables are described in cone search and TableQuery entries in registry.
- Support for ADQL and PDQL but latter is entirely untested
- Support for table uploads, synchronous and asynchronous queries
- Limited support for ADQL geometry constructs
- Java implementation integrated with overall HEASARC data interfaces.
  - Postgres based implementation. No use of proprietary software or tools.

HEASARC Usage

- Almost all usage looks to be testing/ synchronization of data (one query per table per day). Other use at rate of <10-50 queries/day
- Asynchronous usage ~5 queries day
- No usage of PDQL interface (but it is not publicized or even really tested)
- No questions to help desk regarding TAP interfaces from end-users (some from other VO developers)

HEASARC Development Effort

- Major development effort (>1 py) synchronous with the development of the standard
- Modest maintenance effort in past 3 years
  - Bug fixes and security updates
- Implemented as one interface to HEASARC environment
- STILTS TAPlint critical to ensuring reasonable conformance to standard.
  - Testing with validator essential to ensure proper compliance with complex standard

HEASARC Issues

- Implementation of geometry
  - Used implementation based on pgsphere
  - Simple miniscule changes in query syntax (e.g., order of parameters in symmetric functions) can dramatically affect query performance
  - Unclear what is useful minimum geometry support
- Punted on registry metadata for tables and assume users will get that from entries for Cone search services.
  - IRSA facing similar issue

HEASARC Issues

- Security concerns mandated implementation of full SQL parser. Added additional functionality to provide limited geometry support but did not explicitly limit queries to ADQL spec.
- Security monitors do not like direct SQL updates and table metadata downloads.
  - Had to use encoded VOTable format to shut off security alarms on schema downloads.
- Implementation is orders of magnitude more complex than other DAL standards.
- No obvious approach for managing data products associated with tables.
  - ObsTAP tried for one mission
  - Datalink looks promising
- Future of PDQL interface

TAP @ IRSA

- First public deployment in April, will eventually supersede home-grown web API in use for many years
- TAP provides welcome improvement in query expressiveness
- Richness of TAP and ADQL spec has been an impediment to implementation and deployment
  - Difficult to implement full ADQL efficiently
  - Potentially “expensive” in operations as more complex queries can consume more server resources

IRSA: ADQL and Geometry

- ADQL includes geometric features that are beyond the capabilities of IRSA's existing database back-ends (Informix, Oracle, tinyhtm)
- PostgreSQL alternatives not presently realistic for IRSA:
  - IRSA has a number of very large tables (1B-10B rows, some upwards of 40B rows), not seen as a good fit. Some of these rely on structured file storage instead of RDBMS.
  - pgSphere found to be somewhat buggy, not well maintained
  - Even PostGIS and Q3C cannot handle all of ADQL's geometric primitives
- IRSA has implemented a subset of ADQL's geometry support layered on top of existing back-ends

MAST TAP Implementation Status

- Completed very basic Obs-TAP prototype with minimal service features
  1. Science Data Collections
     - Mapping from science catalogs for missions to Obscore using SQL Server DB Views (~30 spectral and image collections)
     - ObsTAP basic search query access with prototype UWS
  2. Preliminary Registry Interface with access to registry DM
- Building enhanced Data Model Infrastructure
  - MAST is developing CAOM-2 as underlying TAP data model across ***ALL*** MAST supported missions (B. McLean)
  - Spatial Regions are indexed with HTM tessellation, STC/s
    - Catalog Source coordinates indexed with HTM IDs (L20),

MAST TAP Service Issues

- Complexity with standard ADQL Regions
  - ADQL region syntax is not a straightforward translation to SQL Server DB - standard more closely aligned with pgSphere format using PostgreSQL
- Initial funding for TAP development was limited to in-kind funding via MAST
  - VAO did not fund archive data center DAL development
  - IVOA Reference Implementation based on JAVA implementations with PostgreSQL

What's Next for MAST TAP

- NAVO project effort for uniform DAL across archive centers
  - Complete a preliminary TAP server full implementation across MAST based on CAOM-2 (later '15)
    - AQDL Parser supporting basic spatial regions
    - UWS enhancements
    - Exercise validators, testing with TAP Clients (TOPCAT, MAST Portal, etc.)
- Complete Registry Interface TAP implementation based on new RegTAP data model
    - Resolve query issues for ADQL/TAP with SQL Server
    - Support IVOA standard RI-2 process
- Publish TAP Services in the STScI NAVO Registry

General concerns

- How do we ensure that TAP usage pays for the considerable development costs? Cost/ Benefit so far has been very high.
- How do we manage the registration of myriad tables in a single TAP services?
- How do we support efficient querying using geometry data?