A TAP Server At The NASA Exoplanet Science Institute

G. Bruce Berriman
John Good
Mihseh Kong (Caltech/IPAC-NExScI)

Why A New TAP Server?

- Lightweight Python implementation (install via pip).
- DBMS agnostic (we use Oracle and SQLite).
- Require no DBMS extensions.

Technical Approach

- The hard work in making the service simple is not TAP itself, but in supporting ADQL.
- Keep the complexity away from the DBMS.
- Instead translate the ADQL into SQL in a way an DBMS can handle.

ADQL to SQL Example

```
where CONTAINS(point('icrs', ra, dec), circle('icrs',
234.56, 34.567, 0.57)) = 1
```

Becomes:

```
where ((-0.47748526711939326*x)+(-
0.67089462951151357*y)+(0.56736955838018899*z)>=9.999505
15308283494e-01)
             AND ((htm7 = 228437) OR (htm7 BETWEEN
228442 AND 228444) OR
                  (htm7 = 228466) OR (htm7 = 228473) OR
(htm7 = 228477) OR (htm7 = 228479))
```

What Is In The Example

- Exact geometric cone constraint and…
- A set of spatial bins to make use of indexing.
- This translation happens before the query is submitted to the DBMS.

Caveats

- Only supports HTM and HPX indexing (any others would have to be added).
- We don't have any databases that contain geometric objects so haven't implemented INTERSECTS().
  - We do have an R-Tree implementation external to the DBMSs that could be folded in.
- Since we place no constraints on the DBMS, we could support uploads and joins but do not have plans to do so.

Deployment in September 2019

- TAP/ADQL code complete. Developing a test plan.
- TAP Schema not yet been fully populated but requires no extra coding.
- The code is maintained in git – will be released on GitHub and uploaded to PyPI. Plan to release with a BSD 3-clause license.
- **If you wish to give it a test drive, get in touch with me at gbb@ipac.caltech.edu**