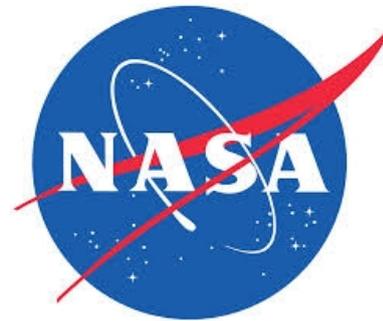


Fine-Grained Table Differences in TAPRegExt

Walter Landry

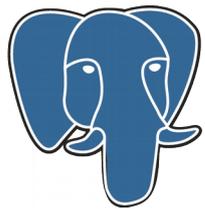


TAP Backends at

- IRSA has a single TAP endpoint with 3 (soon to be 4) different backends.

Informix[®]

ORACLE[®]



PostgreSQL



SQLite

Informix[®] + HTM

- This is our legacy database. We have been slowly migrating our tables to Oracle and hope to finish this soon.
- I do not think that the limitations of this database should be a factor in the long term design of these services.

The Oracle logo consists of the word "ORACLE" in white, uppercase, sans-serif font, set against a red rectangular background. A registered trademark symbol (®) is located at the top right of the letter "E".

+ HTM

- This is where most of our tables are.
- Almost all of the tables (e.g. 2MASS, WISE, IRAS) are spatially indexed using the HTM (Hierarchical Triangular Mesh) tessellation of the sky.
- We translate geometric constraints into HTM ranges plus the actual constraint.
 $((\text{htm} > 100 \text{ AND } \text{htm} < 200)$
 $\text{AND } (x - \text{in}.x)^2 + (y - \text{in}.y)^2$
 $+ (z - \text{in}.z)^2 < r^2)$
 $\text{AND } \{\text{other constraints}\}$

The Oracle logo, consisting of the word "ORACLE" in white, uppercase, sans-serif font, with a registered trademark symbol (®) to the upper right of the letter "E". The logo is set against a solid red rectangular background.

ORACLE[®]

+ HTM

- This allows Oracle to reduce the list of rows to look at when considering the real geometric constraint.
- It does restrict the kind of queries we can handle.

The Oracle logo, consisting of the word "ORACLE" in white, uppercase, sans-serif font, with a registered trademark symbol (®) to the upper right of the "E". The logo is set against a solid red rectangular background.

+ HTM

- However, we do not always require ADQL-style spatial constraints.
- Some of these tables are just not that big (e.g. IRAS), so arbitrary queries can work fine.
- Some of our tables are not spatially organized (e.g. AllWISE Frame Cross-Reference).

HTM + SQLite

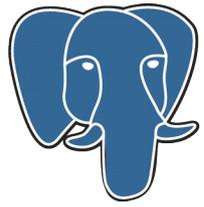
- This is used only for the Planck TOI (Time Ordered Information).
- Each of the 72 detectors from the Planck mission took about 100 measurements/second for about three years. This produced a really, really big table for each detector.
- For fast lookup and smaller files, we store this minimally processed data in a specialized HTM file and use libtinyhtm¹ to access it.



1) <https://github.com/Caltech-IPAC/libtinyhtm/>

HTM + SQLite

- However, libtinyHTM only implements spatial searches.
- So to (partially) implement TAP, we first perform a spatial search, put the results into an in-memory instance of SQLite, and run the rest of the TAP query.
- Because of the size of the tables, geometric constraints are **required**.



PostgreSQL +



- For images, we are standing up a new PostgreSQL + PostGIS backed image table service.
- PostGIS does not handle all of the geometric primitives required by ADQL.

Capability differences

- Some tables require geometry, some encourage it, and some do not allow it.
- Different SQL backends support different capabilities
- Some of this can be smoothed over by rewriting ADQL as needed for each service.

Expressing Fine-Grained Differences in TAPRegExt

- 1) Add table specific elements to TAPRegExt

Expressing Fine-Grained Differences in TAPRegExt

- 1) Add table specific elements to TAPRegExt
 - complicates validating and parsing for everyone

Expressing Fine-Grained Differences in TAPRegExt

- 1) Add table specific elements to TAPRegExt
 - complicates validating and parsing for everyone
- 2) Add per-table examples

Expressing Fine-Grained Differences in TAPRegExt

- 1) Add table specific elements to TAPRegExt
 - complicates validating and parsing for everyone
- 2) Add per-table examples
 - not machine readable

Expressing Fine-Grained Differences in TAPRegExt

- 1) Add table specific elements to TAPRegExt
 - complicates validating and parsing for everyone
- 2) Add per-table examples
 - not machine readable
- 3) Add the metadata you would have added to TAPRegExt to TAP_SCHEMA.tables.

Expressing Fine-Grained Differences in TAPRegExt

- 1) Add table specific elements to TAPRegExt
 - complicates validating and parsing for everyone
- 2) Add per-table examples
 - not machine readable
- 3) Add the metadata you would have added to TAPRegExt to TAP_SCHEMA.tables.
 - needs a way to serialize TAPRegExt