TAPsh

A shell to query database servers in the Virtual Observatory

Ivan Zolotukhin
IRAP, Toulouse

Introduction

- All started as TAP client in GAVO `votable` Python module to test DaCHS
- For quick queries and debug something shell-like is much nicer
- Hence TAPsh as a convenient wrapper over `votable`

Installation

- `virtualenv` is highly recommended
- `easy_install` http://vo.ari.uni-heidelberg.de/soft/dist/tapsh-latest.tar.gz
- Debian systems: install package from GAVO repository
- `java -jar tapsh.jar` (includes Python runtime; good for quick look)

Features

- Auto-completion!
- No sophisticated visualization, hence SAMP
- TAP upload
- TAP services registry

Demo

- Launch tapsh
- Select TAP server
- Connect
- Explore what’s inside
- Make query (sync and async)
- Send to TOPCAT
- Upload table and do JOIN

Useful queries

- Explore TAP service schema

```
SELECT table_name FROM tap_schema.tables
SELECT * FROM tap_schema.columns WHERE table_name = 'public.ident'
```

- Upload with TAPsh

```
SELECT * FROM tap_upload.qq
upload "test.vot" as qq
run
dump
```

- Async queries with TAPsh

```
SELECT COUNT(1) FROM ucac4.main
start
job
dump
```

Useful queries

- Cone search in SDSS hosted at VizieR

```
SELECT
  *
FROM
  "II/294/sdss7"
WHERE
  1=CONTAINS(POINT('ICRS', RAJ2000, DEJ2000), CIRCLE('ICRS', 189.2, 62.21, 0.5))
  AND zsp > 0.001
  AND zsp < 0.2
```

- Simbad

```
SELECT
  main_id, mrot.vsini, mrot.bibcode
FROM
  basic
JOIN
  ident AS i ON i.oidref = oid
JOIN
  mesRot AS mrot ON mrot.oidref = oid
WHERE
  i.id IN ('*  51 Peg', 'V* bet Cep')
  AND mrot.bibcode >= '2000'
```

Conclusions

- Good for quick tests
- Good for those who are used to SQL shells
- Good for relational registry queries
- Good for quick check of TAP services
- Hence somewhat limited to geeks

Recommended links

- Homepage: http://vo.ari.uni-heidelberg.de/soft/tapsh
- ADQL course by GAVO: http://docs.g-vo.org/adql/html/
- TAP / ADQL cheat sheet by GAVO: http://docs.g-vo.org/adqlref/adqlref.pdf

Questions?

Python libraries for TAP

Something that doesn’t quite exist

Ivan Zolotukhin
IRAP, Toulouse

Quick and dirty: do it yourself

- Works for 80% of real life problems: sync queries

```
import urllib2

accessURL = "http://tapvizier.u-strasbg.fr/TAPVizieR/tap"
query = "SELECT TOP 3 * FROM TAP_SCHEMA.tables"

query_url = '%s/sync?REQUEST=doQuery&LANG=ADQL&QUERY=%s' % (accessURL,
urllib2.quote(query))

print query_url
```

`votable` by GAVO

```python
from gavo import votable

accessURL = "http://tapvizier.u-strasbg.fr/TAPVizieR/tap"
query = "SELECT TOP 3 * FROM TAP_SCHEMA.tables"

job = votable.ADQLTAPJob(accessURL, query)
job.run()

dataIterator, metadata = votable.load(job.openResult())
job.delete()

print list(dataIterator)
```

`votable` by GAVO

- Full featured implementation: async, uploads, jobs, etc.
- Not very user friendly
- OK for developers
- Docs: http://vo.ari.uni-heidelberg.de/docs/DaCHS/tapquery.html

`pyvo` by Ray Plante & Co

- Provides SIA, SSA, SCS, SLA access
- Aims to have TAP client (but no progress since 2 years)
- Aims to integrate to `astropy`
- I think the real progress will happen here: https://github.com/astropy/astroquery/issues/243 (raised by F. Paletou and myself: Simbad access with `astropy` is limited without TAP) and https://github.com/astropy/astroquery/issues/277 (follow-up on TAP capabilities in `astropy`)

Conclusions

- Not a huge choice
- When `astropy` / `astroquery` implements it, TAP user base will increase
- For now usually easier to do by hand

Few points on TAP

- Cumbersome and difficult to use
- **...but provides extraordinary research opportunities**
- Need to involve researchers and data centers to use it
- Then synergy and great profits: see SDSS CasJobs, most successful research project in history

Questions?