

TOPCAT and STILTS

Mark Taylor (Bristol)

AIDA Fifth Technology Forum

16 March 2010

`$Id: tcetc.tex,v 1.5 2010/03/16 13:48:28 mbt Exp $`

Outline

- Recent Enhancements (*releases Nov, Dec 2009*)
- Crossmatching (*work in progress*)
- JyStilts (*work in progress*)
- Registry use (*experience and progress*)

Recent Enhancements

- Released November 2009 (just before ESO interop):
 - Multiple table load
 - VOTable 1.2
 - Vizier load dialogue (*with help from CDS*)
 - Single DAL query (Cone, SIA, SSA) window improvements
 - Multiple DAL query (Cone, SIA, SSA) window added
 - Spectrum display now via SAMP, not internal SPLAT
 - VO menu
 - Adaptive storage policy by default
 - . . . and others
- Released December 2009:
 - Graphics export to PDF (using [iText](#))

Crossmatching

- Aim to improve crossmatch scalability
 - Current limits: few $\times \sim 10^6$ rows? limited by memory
 - Target: few $\times \sim 10^7$ rows? (much bigger is a different problem)
 - Users starting to hit this limit
 - Various approaches:
 - ▷ multi-stage bin/match cycle
 - ▷ optimise bin size
 - ▷ micro-tweaking existing algorithms/data structures
 - ▷ provide user tuning options
 - ▷ modify algorithms to use (optionally) disk-based storage
 - Many different regimes of matching; difficult to optimise for all at once
 - Significant efficiency improvements achieved so far for example matches
 - ▷ memory usage: factor of ~ 5
 - ▷ CPU time: factory of ~ 3

Jython interface to STILTS scripting

- Prompted by a HIPE user (Herschel Interactive Processing Environment - Jython based)
- Jython is a pure-java implementation of Python language:
 - ▷ Language is identical to CPython
 - ▷ . . . but no C- or FORTRAN-based extensions (e.g. Numpy, Scipy)
 - ▷ Access to java classes/objects is very straightforward
- Advantages
 - ▷ Powerful and familiar environment for STILTS scripts (better than `/bin/csh`)
 - ▷ Syntax can be less convoluted (especially string functions)
 - ▷ Intermediate results stored in JVM can eliminate serialize/deserialize round trips:
 - Long or complex scripts much more efficient
 - Integration into existing table-processing contexts much more efficient
- Easy to implement
 - ▷ single script `stilts.py`, autogenerated at build time
 - ▷ up and running in a couple of days
 - ▷ complete (with pythonic syntax elements, docs, . . .) in a couple of weeks
 - ▷ helped by the fact that STILTS commands/params are self-describing objects
- Currently available as [pre-release](#) (comments from python natives solicited)

STILTS/JyStilts Comparison

Command-line:

```
stilts tskymatch2 in1=survey.fits \  
                icmd1='addskycoords fk4 fk5 RA1950 DEC1950 RA2000 DEC2000' \  
                in2=mycat.csv ifmt2=csv \  
                icmd2='select VMAG>18' \  
                ra1=ALPHA dec1=DELTA ra2=RA2000 dec2=DEC2000 \  
                error=10 join=2not1 \  
                out=matched.fits
```

Jython:

```
>>> import stilts  
>>>  
>>> t1 = stilts.tread('survey.fits')  
>>> t1 = t1.cmd_addskycoords(t1, 'fk4', 'fk5', 'RA1950', 'DEC1950', 'RA2000', 'DEC2000')  
>>>  
>>> t2 = tread('mycat.csv', 'csv')  
>>> t2 = t2.cmd_select('VMAG>18')  
>>>  
>>> tm = skymatch2(in1=t1, in2=t2, ra1='ALPHA', dec1='DELTA', error=10, join='2not1')  
>>>  
>>> tm.write('matched.fits')
```

JyStilts Examples

● Interactivity

```
>>> tm.mode_count()
columns: 19  rows: 2102
>>> tm.cmd_keepcols('ID ALPHA DELTA').cmd_head(2).write()
+-----+-----+-----+
| ID      | ALPHA          | DELTA          |
+-----+-----+-----+
| 262     | 149.82439      | -0.11249      |
| 263     | 150.14438      | -0.11785      |
+-----+-----+-----+
```

● Python syntax for table objects

- Iterable: `for row in table: ...`
- Subscriptable (if random access): `table[5], table[5:10]`
- Add/multiply for concatenation: `table1 + table2, table1 * 100`
- Column/cell access by name/index: `table[1000]['BMAG'], table.coldata(4)[1000]`
- Column metadata: `table.columns()[2].getUCD()`
- Table metadata: `table.parameters()['DATE']`
- . . . but data access is much faster using STILTS methods

Registry

TOPCAT registry queries

- Needs list of Cone, SIA, SSA services
- Use RI 1.0 interface for compatibility with all VO registries
 - ▷ Available SOAP operations:
 - **Search**: free-form ADQL query
 - **KeywordSearch**: search on keywords *no other constraints permitted*
 - **XQuerySearch**: free-form XQuery *optional in RI 1.0*
 - ▷ Only **Search** is suitable

The screenshot shows a web interface for 'Cone Search'. The 'Registry' dropdown is set to 'http://registry.astrogrid.org/astrogrid-registry/services/RegistryQueryv1_0'. The 'Keywords' field contains 'SDSS QSO'. Below the search fields is a table of results with columns 'shortName', 'title', and 'AccessURL'. The table lists several SDSS-related services, with 'SDSSXMMQSO' selected. Below the table is a 'Cone Search Parameters' section with a 'Cone Search URL' field containing 'http://heasarc.gsfc.nasa.gov/cgi-bin/vo/cone/coneGet.pl?table=sdssxmmqso'.

shortName	title	AccessURL
J/ApJ/639/766	SDSS-DR3 strong MgII absorbers (Prochter+, 2006)	CDS suppo
VII/252	SDSS-DR5 quasar catalog (Schneider+, 2007)	CDS suppo
J/ApJ/166/470	SDSS-Spitzer type I QSOs IR photometry (Richards+, 2006)	CDS suppo
J/MNRAS/386/1252	SWIRE/SDSS quasars (Hatziminaoglou+, 2008)	CDS suppo
SDSSXMMQSO	Sloan Digital Sky Survey (DR5)/XMM-Newton Quasar Survey...	Michael Pre
SDSSBALQSO	Sloan Digital Sky Survey Broad Absorption Line Quasars Cat...	Michael Pre
SDSSBALQSO2	Sloan Digital Sky Survey Broad Absorption Line Quasars Cat...	Michael Pre
SDSSCXOQSO	Sloan Digital Sky Survey Quasars Detected by Chandra	Michael Pre

Registry Search Query

Search SOAP query for services related to 2MASS

- ADQL/S:

```
Search WHERE ( capability/@standardID = 'ivo://ivoa.net/std/ConeSearch' ) AND
( (identifier LIKE '%2MASS%' OR
  content/description LIKE '%2MASS%' OR
  title LIKE '%2MASS%' OR
  content/subject LIKE '%2MASS%' OR
  content/type LIKE '%2MASS%') )
```

Registry Search Query

Search SOAP query for services related to 2MASS

- ADQL/X:

```
<Where xmlns:ad="http://www.ivoa.net/xml/ADQL/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ad:Condition xsi:type="ad:intersectionSearchType">
    <ad:Condition xsi:type="ad:closedSearchType">
      <ad:Condition xsi:type="ad:comparisonPredType" Comparison="">
        <ad:Arg xsi:type="ad:columnReferenceType" Table="" xpathName="capability/@standardID" name="@standardID">
          <ad:Arg xsi:type="ad:atomType">
            <ad:Literal xsi:type="ad:stringType" Value="ivo://ivoa.net/std/ConeSearch"/>
          </ad:Arg>
        </ad:Condition>
      </ad:Condition>
    <ad:Condition xsi:type="ad:closedSearchType">
      <ad:Condition xsi:type="ad:closedSearchType">
        <ad:Condition xsi:type="ad:unionSearchType">
          <ad:Condition xsi:type="ad:unionSearchType">
            <ad:Condition xsi:type="ad:unionSearchType">
              <ad:Condition xsi:type="ad:unionSearchType">
                <ad:Condition xsi:type="ad:likePredType">
                  <ad:Arg xsi:type="ad:columnReferenceType" Table="" xpathName="identifier" name="identifier"/>
                  <ad:Pattern xsi:type="ad:atomType">
                    <ad:Literal xsi:type="ad:stringType" Value="%2MASS%"/>
                  </ad:Pattern>
                </ad:Condition>
              <ad:Condition xsi:type="ad:likePredType">
                <ad:Arg xsi:type="ad:columnReferenceType" Table="" xpathName="content/description" name="descr">
                  <ad:Pattern xsi:type="ad:atomType">
                    <ad:Literal xsi:type="ad:stringType" Value="%2MASS%"/>
                  </ad:Pattern>
                </ad:Condition>
              </ad:Condition>
            <ad:Condition xsi:type="ad:likePredType">
              <ad:Arg xsi:type="ad:columnReferenceType" Table="" xpathName="title" name="title"/>
              <ad:Pattern xsi:type="ad:atomType">
                <ad:Literal xsi:type="ad:stringType" Value="%2MASS%"/>
              </ad:Pattern>
            </ad:Condition>
          </ad:Condition>
        </ad:Condition>
      </ad:Condition>
    </ad:Condition>
  </ad:Condition>
  ....

```

Registry Use: Libraries

- Used Ray Plante's [IVOARegistry](#) library
 - Mostly good
 - ▷ takes care of ADQL/S → ADQL/X — *phew*
 - ▷ one or two small bugs, easy to fix or work around
 - Builds DOM for registry response
 - ▷ Memory hungry for large search responses
 - ▷ Can break up request into several sections — but how big?
 - ▷ Some single resource records are very large (e.g. UKIDSS at WFAU 1Mb XML → several Mb DOM per resource)
- Ended up writing custom lightweight registry SOAP client
 - Uses IVOARegistry lib for ADQL/S→ADQL/X translation
 - Uses small DIY SOAP client (Axis not required - 500 lines is better than 200k)
 - Extracts required resource info using SAX
 - Fast, very low memory footprint
 - Could be packaged for 3rd party use — any takers?

Registry Use: Other issues

- Problems using mandatory parts of RI 1.0 standard:
 - Several registries not initially compliant with standard
 - Not obvious how to write “*give me all cone searches*” query
 - Can write “*give me all searchable registries*” — but results are wrong
 - ADQL search by search term is/isn't/may be case-sensitive
- Is anybody else using this interface?

Mailing Lists

Belatedly introduced TOPCAT/STILTS/STIL mailing lists:

`topcat-user@bristol.ac.uk`

`topcat-announce@bristol.ac.uk`

▶ see web site for subscription details