CDS X-match service API

François-Xavier Pineau[1], Thomas Boch[1]

[1]CDS, Observatoire Astronomique de Strasbourg

IVOA Interop, Heidelberg

The CDS X-match HTTP API: Introduction

Description

- One URL (GET and POST) to synchronously perform a X-match between:
  - ▸ a VizieR table
  - ▸ a flat view of Simbad
  - ▸ an uploaded list of objects (file or URL)
- Several URLs to retrieve metadata:
  - ▸ list of available tables
  - ▸ columns available for a given table
- Queriable manually or through softwares => easily scriptable

Motivation

- Provide a simple scriptable synchronous X-match for "small" jobs
- Reduce the VizieR load
  - ▸ X-match by millions of (multi-threaded) cone searches on VizieR
    - Inefficient way to perform a X-match
    - Useless overload of the servers

The CDS X-match HTTP API: Introduction

Description

- One URL (GET and POST) to synchronously perform a X-match between:
  - a VizieR table
  - a flat view of Simbad
  - an uploaded list of objects (file or URL)
- Several URLs to retrieve metadata:
  - list of available tables
  - columns available for a given table
- Queriable manually or through softwares $=>$ easily scriptable

Motivation

- Provide a simple scriptable synchronous X-match for "small" jobs
- Reduce the VizieR load
  - X-match by millions of (multi-threaded) cone searches on VizieR
    - ⋆ Inefficient way to perform a X-match
    - ⋆ Useless overload of the servers

Use of the IVOA Data Access Layer Interface (DALI)

Parts implemented

- Input parameters: (? means optional)
  - **request**=xmatch
  - cat1|2=NAME|URL|FILE (NAME = simbad|vizier:II/246/out)
  - ? colRA1|2=STRING
  - ? colDec1|2=STRING
  - distMaxArcsec=DOUBLE
  - ? selection=best|all
  - **responseformat**=CSV|**VOTABLE**|JSON
  - ? cols1|2=STRING,STRING,STRING
  - **?** **maxrec**=INT
- VOTable informations: **QUERY_STATUS**, **OVERFLOW**
- Resources: **availability**, **capabilities**, tables

Not (yet?) implemented

- Parameters: **upload** (simpler without it), **version** (optional), **runid** (optional)
- Resources: **examples** (optional)

Use of the IVOA Data Access Layer Interface (DALI)

Parts implemented

- Input parameters: (? means optional)
  - **request**=xmatch
  - cat1|2=NAME|URL|FILE (NAME = simbad|vizier:II/246/out)
  - ? colRA1|2=STRING
  - ? colDec1|2=STRING
  - distMaxArcsec=DOUBLE
  - ? selection=best|all
  - **responseformat**=CSV|**VOTABLE**|JSON
  - ? cols1|2=STRING,STRING,STRING
  - **?** **maxrec**=INT
- VOTable informations: **QUERY_STATUS**, **OVERFLOW**
- Resources: **availability**, **capabilities**, tables

Not (yet?) implemented

- Parameters: **upload** (simpler without it), **version** (optional), **runid** (optional)
- Resources: **examples** (optional)

Implementation

Same engine as the asynchronous X-match service:

- HEALPix partitioning (tables > a few millions rows)
- On-the-fly multi-threaded creation of kd-trees on α, δ
- 2 steps:
  1. Compute links (a M:N relationship table j, including angular separation)
  2. On-the-fly generation of the output file (2 joins: $table_1 \times j \times table_2$)

Performances

- Often very fast: (see *computationTime* in the returned VOTable)
  1. M:N relation computed in less than a few seconds
  2. then bottleneck is your internet connection
  3. Examples:
     - Hipparcos vs Tycho2 (118 k vs 2.5 M): computation time $\sim 0.5\,s$
     - 2dFGRS vs USNO-A.2 (404 k vs 526 M) : computation time $\sim 9\,s$
- Worst case: a small all-sky catalog VS a large all-sky[1] catalog
  - e.g. I/239/hip_main (100 k sources) vs SDSS DR8 (500 M sources)
  - bottleneck is HDD seek time[2] (if data not in cache)
  - could be improved ($\times 10 - 20$) using SSDs

Implementation

Same engine as the asynchronous X-match service:

- HEALPix partitioning (tables > a few millions rows)
- On-the-fly multi-threaded creation of kd-trees on α, δ
- 2 steps:
  1. Compute links (a M:N relationship table j, including angular separation)
  2. On-the-fly generation of the output file (2 joins: $table_1 \times j \times table_2$)

Performances

- Often very fast: (see *computationTime* in the returned VOTable)
  1. M:N relation computed in less than a few seconds
  2. then bottleneck is your internet connection
  3. Examples:
     - ⋆ Hipparcos vs Tycho2 (118 k vs 2.5 M): computation time $\sim 0.5\,s$
     - ⋆ 2dFGRS vs USNO-A.2 (404 k vs 526 M) : computation time $\sim 9\,s$
- Worst case: a small all-sky catalog VS a large all-sky[1] catalog
  - ▸ e.g. I/239/hip_main (100 k sources) vs SDSS DR8 (500 M sources)
  - ▸ bottleneck is HDD seek time[2] (if data not in cache)
  - ▸ could be improved ($\times 10 - 20$) using SSDs

Options & Limits

- Input parameters: (? means optional)
    - **request**=xmatch
    - cat1|2=NAME|URL|FILE (max = 100 MB, NAME = simbad|vizier:II/246/out)
    - ? colRA1|2=STRING
    - ? colDec1|2=STRING
    - distMaxArcsec=DOUBLE (value max = 180)
    - ? selection=best|all (default = all)
    - **response**format=CSV|**VOTABLE**|JSON
    - ? cols1|2=STRING,STRING,...,STRING
    - ? **maxrec**=INT (value max = 2 000 000)

Other limitations

- Output limited to 2 000 000 rows, OVERFLOW info (VOTable) if more
- For VizieR tables, column choice limited to VizieR default columns
- Max 5 jobs at the same time

Options & Limits

- Input parameters: (? means optional)
  - **request**=xmatch
  - cat1|2=NAME|URL|FILE (max = 100 MB, NAME = simbad|vizier:II/246/out)
  - ? colRA1|2=STRING
  - ? colDec1|2=STRING
  - distMaxArcsec=DOUBLE (value max = 180)
  - ? selection=best|all (default = all)
  - **responseformat**=CSV|**VOTABLE**|JSON
  - ? cols1|2=STRING,STRING,...,STRING
  - ? **maxrec**=INT (value max = 2 000 000)

Other limitations

- Output limited to 2 000 000 rows, OVERFLOW info (VOTable) if more
- For VizieR tables, column choice limited to VizieR default columns
- Max 5 jobs at the same time

Example

Directly in a web browser

http://cdsxmatch.u-strasbg.fr/xmatch/api/v1/sync?**REQUEST=xmatch**
&**cat1**=vizier:V/123A/cv &**cat2**=vizier:I/239/hip_main &**distMaxArcsec**=3
&**RESPONSEFORMAT**=votable

- VizieR table 1: V/123A/cv (Catalog of CVs, Downes et al. 2006)
- VizieR table 2: I/239/hip_main (Hipparcos main catalogues, ESA 1997)
- X-match distance: 3 arcsec
- Output format: VOTable

Or with wget

wget -O downes_vs_hip_3arcsec.vot
'http://cdsxmatch.u-strasbg.fr/xmatch/api/v1/sync?**REQUEST=xmatch**
&**cat1**=vizier:V/123A/cv &**cat2**=vizier:I/239/hip_main &**distMaxArcsec**=3
&**RESPONSEFORMAT**=votable'

Example

Directly in a web browser

http://cdsxmatch.u-strasbg.fr/xmatch/api/v1/sync?**REQUEST=xmatch**
&**cat1**=vizier:V/123A/cv &**cat2**=vizier:I/239/hip_main &**distMaxArcsec**=3
&**RESPONSEFORMAT**=votable

- VizieR table 1: V/123A/cv (Catalog of CVs, Downes et al. 2006)
- VizieR table 2: I/239/hip_main (Hipparcos main catalogues, ESA 1997)
- X-match distance: 3 arcsec
- Output format: VOTable

Or with **wget**

wget -O downes_vs_hip_3arcsec.vot
'http://cdsxmatch.u-strasbg.fr/xmatch/api/v1/sync?**REQUEST=xmatch**
&**cat1**=vizier:V/123A/cv &**cat2**=vizier:I/239/hip_main &**distMaxArcsec**=3
&**RESPONSEFORMAT**=votable'

Example

Using **curl** to match several FITS file with Simbad in Bash

```
for f in file1 file2 file3 file4; do \
    curl -X POST -F request=xmatch \
        -F cat1=@$f.fits -F colRA1=RAJ2000 -F colDec1=DEJ2000 \
        -F cat2=simbad \
        -F distMaxArcsec=25 \
        -F RESPONSEFORMAT=csv \
        http://cdsxmatch.u-strasbg.fr/xmatch/api/v1/sync \
        > $f_vs_simbad_25arcsec.csv \
done
```

Other languages

For Python, Ruby and Java, see here:
http://cdsxmatch.u-strasbg.fr/xmatch/doc/xmatch-API-usage-examples.html

Example

Using **curl** to match several FITS file with Simbad in Bash

```
for f in file1 file2 file3 file4; do \
    curl -X POST -F request=xmatch \
        -F cat1=@$f.fits -F colRA1=RAJ2000 -F colDec1=DEJ2000 \
        -F cat2=simbad \
        -F distMaxArcsec=25 \
        -F RESPONSEFORMAT=csv \
        http://cdsxmatch.u-strasbg.fr/xmatch/api/v1/sync \
        > $f_vs_simbad_25arcsec.csv \
done
```

Other languages

For Python, Ruby and Java, see here:
`http://cdsxmatch.u-strasbg.fr/xmatch/doc/xmatch-API-usage-examples.html`

Resource **tables**

- Does not follow the VODataService standard!
- http://cdsxmatch.u-strasbg.fr/xmatch/api/v1/sync/tables
  - ▸ the list of all availables tables + some informations (JSON format)
- http://cdsxmatch.u-strasbg.fr/xmatch/api/v1/sync/tables?
  - ▸ action=**getVizieRTableNames**&RESPONSEFORMAT=json|txt
    - ★ returns the list of VizieR tables available
  - ▸ action=**getColList**&tabName=TABLE_NAME&RESPONSEFORMAT=json|txt
    - ★ returns the list of the column available for the given table
  - ▸ action=**getAliases**
    - ★ returns the list of aliases available (for large catalogs, JSON format)
  - ▸ action=**getInfo**&tabName=TABLE_NAME
    - ★ returns some informations about a VizieR table (JSON format)

Base URL:
http://cdsxmatch.u-strasbg.fr/xmatch/api/v1/sync

Documentation:
http://cdsxmatch.u-strasbg.fr/xmatch/doc/cross-match-API.html

Contact us for any trouble, bug, suggestion, ...

- francois-xavier.pineau@astro.unistra.fr
- thomas.boch@astro.unistra.fr

THANK YOU

Base URL:
http://cdsxmatch.u-strasbg.fr/xmatch/api/v1/sync

Documentation:
http://cdsxmatch.u-strasbg.fr/xmatch/doc/cross-match-API.html

Contact us for any trouble, bug, suggestion, ...

- francois-xavier.pineau@astro.unistra.fr
- thomas.boch@astro.unistra.fr

THANK YOU

Base URL:
http://cdsxmatch.u-strasbg.fr/xmatch/api/v1/sync

Documentation:
http://cdsxmatch.u-strasbg.fr/xmatch/doc/cross-match-API.html

Contact us for any trouble, bug, suggestion, ...

- francois-xavier.pineau@astro.unistra.fr
- thomas.boch@astro.unistra.fr

THANK YOU