Astronomical Data Analysis Software and Systems XXI
ASP Conference Series, Vol. 461
Pascal Ballester, Daniel Egret, and Nuria P. F. Lorente, eds.

Astronomical Data Processing Using SciQL, an SQL Based Query Language for Array Data

Ying Zhang,[1] Bart Scheers,[1,2] Martin Kersten,[1] Milena Ivanova,[1] and Niels Nes[1]

[1]*Centrum Wiskunde & Informatica (CWI), Science Park 123, Amsterdam, The Netherlands*
[2]*Astronomical Institute "Anton Pannekoek", University of Amsterdam, Science Park 904, Amsterdam, The Netherlands*

Abstract. SciQL (pronounced as 'cycle') is a novel SQL-based array query language for scientific applications with both tables and arrays as first class citizens. SciQL lowers the entrance fee of adopting relational DBMS (RDBMS) in scientific domains, because it includes functionality often only found in mathematics software packages. In this paper, we demonstrate the usefulness of SciQL for astronomical data processing using examples from the Transient Key Project of the LOFAR radio telescope. In particular, how the LOFAR light-curve database of all detected sources can be constructed, by correlating sources across the spatial, frequency, time and polarisation domains.

1. Introduction

The ever growing use of high precision experimental instruments in astronomy, e.g., SDSS, LSST, Pan-STARRS and LOFAR, amounts to an avalanche of data to be stored, curated and analysed. Ingestion of terabytes of data on a daily basis is happening in many projects. Planned experimental devices are expected to scale ingestion up to petabytes soon. Efficient data management as part of a data exploration infrastructure has become a discriminative factor for scientific progress. RDBMSs are the prime means to fulfill the role of application mediator for data exchange and data persistence.

Nevertheless, scientific applications are still poorly served by contemporary relational DBMSs. At best, the system provides a bridge towards an external library using user-defined functions, explicit import/export facilities or linked-in Java/C# interpreters. To bridge the gap between the needs of the data-intensive scientific research fields like astronomy and the current DBMS technologies, we have introduced SciQL (Kersten et al. 2011; Zhang et al. 2011), a novel SQL-based array query language for scientific applications with both tables and arrays as first class citizens. SciQL provides a seamless symbiosis of array-, set-, and sequence- interpretation. A key innovation is the extension of value-based grouping in SQL:2003 with structural grouping, i.e., fixed-sized and unbounded groups based on explicit relationships between the dimensional attributes of array cells. This leads to a generalisation of window-based query processing with wide applicability in science domains, e.g., Fourier Transforming light curves, Principle Component Analysis, moving averages, correlation and convolution.

In this paper, we demonstrate the effectiveness of SciQL for astronomical data processing with examples from the Transients Key Project (TKP) of LOFAR.[1] The TKP focuses on studying the explosive and dynamic universe by observing transient and variable radio sources in the frequency ranges 30 – 80 MHz and 120 – 240 MHz. One of the main goals of TKP is building a full Stokes spectral light-curve database of all detected sources. Therefore, association of sources across frequency bands (including external catalogues), Stokes parameters and timestamps is essential. Tens of GBs of extracted data per day needs to be stored in the light-curve database (Scheers 2011). For traditional RDBMSs, the necessary array oriented operations are extremely hard to express in SQL and optimise for query execution. With SciQL, however, such operations can be expressed easily and concisely. Moreover, by revealing the properties of array data, SciQL opens up plenty of opportunities to enhance the data mining possibilities for real-time transient and variability searches.

2. Modelling TKP Data

This section shows how the TKP data is stored in SciQL arrays. Information derivation is discussed in Section 3.

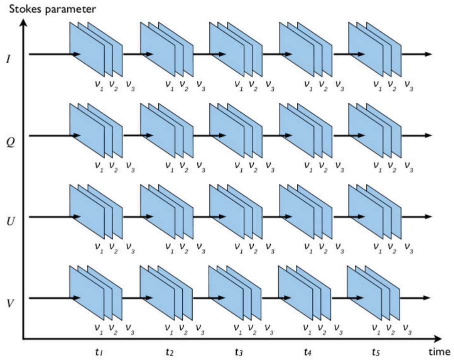

Figure 1. Schematic view of the TKP pipeline input streams of image cubes produced by one observation.

Catalogued Sources. The LOFAR imaging pipeline produces calibrated images, characterised by, e.g., frequency, Stokes parameter and observation timestamp. The data set at each Stokes parameter can be seen as a stream of image cubes arriving at subsequent timestamps, as depicted in Figure 1. An image cube has the same observational timestamp, while individual image planes in the cube fall in different frequency bands. In the LOFAR pipeline, algorithms are applied to extract sources from the images and associate them with earlier detected LOFAR sources across frequency bands, times and Stokes parameters. The LOFAR catalogue contains all measured properties of the detected sources, which are stored in the array `LOFARsrc`:

```
CREATE ARRAY LOFARsrc (mrdn INT DIMENSION[0:1:361], zone INT DIMENSION[-90:1:91],
  ts TIMESTAMP DIMENSION, freq INT DIMENSION[30:10:241], id INT DIMENSION[0:1:*],
  stks CHAR(1) DIMENSION CHECK (stks = ‘I’ OR stks = ‘Q’ OR stks = ‘U’ OR stks = ‘V’),
  ra DOUBLE, decl DOUBLE, ra_err DOUBLE, decl_err DOUBLE, flux DOUBLE, ...);
```

Since the astronomers often search for sources in a certain area in the sphere or sources nearby, we divide the sphere into a fixed number of areas by defining two *dimensional attributes* (for short: *dimensions*): `mrdn` and `zone`, This is inspired by the zone algorithm of Gray et al. (2006), but we also divide the sphere along the meridian (`mrdn`) to create smaller areas. Auxiliary values of the sources are stored as *non-dimensional attributes* (for short: *cell values*), e.g., the `id` of each unique source, the polar coordinates `(ra, decl)` and their errors `(ra_err, decl_err)`, and the `flux`.

All sources from external catalogues are stored in the array `ExtCatSrc` below, which only differs from `LOFARsrc` with two more columns. The dimension `catnm` distinguishes sources from different catalogues. The `orig_id` column contains source IDs

[1] www.lofar.org/astronomy/transients-ksp/transients-key-science-project

taken from their originating catalogues, while the `id` dimension contains newly computed source IDs that are unique in this array. Currently, sources from VLSS, WENSS and NVSS are used. These catalogues do not trace the sources over time and they only measure the total flux intensity at one frequency. Thus, sources from these catalogues always have `‘I’` for `stks`, 0 for `ts`, and the frequency of the catalogue in MHz for `freq`, i.e., 74, 325 and 1400, respectively.

```
CREATE ARRAY ExtCatSrc (LIKE LOFARsrc, orig_id INT,
  catnm VARCHAR(10) DIMENSION CHECK (catnm = ‘NVSS’ OR catnm = ‘VLSS’ OR catnm = ‘WENSS’));
```

Associated Sources. An important operation in the TKP pipeline is to cross-correlate a LOFAR source with known sources in the major external catalogues. This way one can keep track of sources and fluxes at positions of interest in the sky. All information of credible associations are stored in the array `AssocSrc`:

```
CREATE ARRAY AssocSrc (lofar INT DIMENSION[0:1:*], vlss INT DIMENSION[0:1:*],
  wenss INT DIMENSION[0:1:*], nvss INT DIMENSION[0:1:*], w_dist DOUBLE, s_idx DOUBLE, ...);
```

A dimension is defined for each catalogue to contain source IDs. A dimension value 0 means that the corresponding catalogue is excluded when computing the auxiliary values of a cell, which usage will be shown in the example below. All auxiliary values quantifying an association is stored as cell values, e.g., the weighted distance `w_dist` and the spectral index `s_idx`. An empty cell, i.e., all its values are `NULL`, means that it is unknown yet if the sources identified by this cell can be associated. A `w_dist` of -1 denotes that the sources identified by the cell's dimensions cannot be associated.

LOFAR	VLSS	WENSS	NVSS
11	89	-	21

Assume the following associations: , i.e., the LOFAR source 11 is associated with the VLSS source 89 and NVSS source 21, thus the array cells `AS[11][89][0][0]` and `AS[11][0][0][21]` are filled with auxiliary values. Since no association is found in WENSS, all cells `AS[11][89][1:*][21]` have `w_dist` $= -1$.

3. Querying TKP Data

Cross-Correlating Multiple Catalogues. The TKP pipeline matches sources using three association parameters. For simplicity, we only use the weighted distance, defined as: $r = \sqrt{(\Delta\alpha)^2/\sigma^2_{\Delta\alpha} + (\Delta\delta)^2/\sigma^2_{\Delta\delta}}$ with $\Delta\alpha = \alpha_i \cos(\delta_i) - \alpha_j \cos(\delta_j)$ and $\sigma^2_{\Delta\alpha} = \sigma^2_{\alpha_i} + \sigma^2_{\alpha_j}$. In the arrays, the values of α, δ, σ_α and σ_δ are stored as `ra`, `decl`, `ra_err` and `decl_err`, respectively. For every LOFAR source, the SciQL query below searches in each external catalogue to find credible associations by checking their weighted distance.

```
INSERT INTO AssocSrc SELECT L.id, E[*][*][74][*][‘VLSS’].orig_id,
  E[*][*][325][*][‘WENSS’].orig_id, E[*][*][1400][*][‘NVSS’].orig_id,
  SQRT(POWER((AVG(L.ra)*COS(AVG(L.decl)) - E.ra*COS(E.decl)), 2) /
    (POWER(AVG(L.ra_err), 2) + POWER(E.ra_err, 2)) + POWER((AVG(L.decl) - E.decl), 2) /
      (POWER(AVG(L.decl_err), 2) + POWER(E.decl_err, 2))) AS w_dist,
  LOG((AVG(L.flux) / E.flux) / (AVG(L.freq) / E.freq)) AS s_idx
FROM LOFARsrc[*][*][*][*][*][‘I’] AS L, ExtCatSrc[*][*][0][*][*][‘I’] AS E
GROUP BY L.id, E[L.mrdn-1:L.mrdn+2][L.zone-1:L.zone+2], E.id HAVING w_dist < @r_max;
```

First, the query uses *array slicing* (Zhang et al. 2011) in the `FROM` clause to extract only the Stokes `‘I’` from both arrays and the timestamp 0 from `ExtCatSrc`. Then, for every LOFAR source, a group is constructed with every nearby external sources. The second `GROUP BY` condition uses the *array tiling* technique. A credible association is inserted into `AssocSrc` together with the auxiliary values `w_dist` and `s_idx`.

Full Stokes Spectral Light Curves. After all arrays (i.e., sources + associations) are filled with data, we can query them to produce various plots. For instance, the query below builds a spectrum of Stokes `‘I’` of the LOFAR source 11:

```
SELECT * FROM (SELECT freq, AVG(flux) AS flux FROM AssocSrc[11] AS A,
  LOFARsrc[*][*][*][*][11][‘I’] AS L GROUP BY L.ts UNION
  SELECT freq, flux FROM AssocSrc[11] AS A, ExtCatSrc[*][*][0][*][*][‘I’] AS E
  WHERE E[*][*][74][*][‘VLSS’].orig_id = A.vlss
     OR E[*][*][325][*][‘WENSS’].orig_id = A.wenss
     OR E[*][*][1400][*][‘NVSS’].orig_id = A.nvss) AS Spectrum ORDER BY freq;
```

With the information stored in AssocSrc, one can analyse how the flux intensity of a source in an existing catalogue behaves over time in the LOFAR frequency bands. Consider the example in Section 2, in which the LOFAR source 11 is associated with the NVSS source 21. Assume one is interested in the similarity of the flux of the NVSS source 21 over time in the LOFAR frequency bands 30 MHz and 200 MHz. This requires computing the cross-correlation of the two time-series at these frequency bands. With SciQL, it is easy to express an operation like cross-correlation directly in the query:

```
DECLARE fcnt INT, gcnt INT;
SET fcnt = SELECT COUNT(*) FROM LOFARsrc[*][*][*][30][11][‘I’];
SET gcnt = SELECT COUNT(*) FROM LOFARsrc[*][*][*][200][11][‘I’];

CREATE ARRAY VIEW F (idx INT DIMENSION[0:1:fcnt], flux DOUBLE DEFAULT 0.0) AS
  SELECT flux FROM LOFARsrc[*][*][*][30][11][‘I’];
CREATE ARRAY VIEW G (idx INT DIMENSION[0:1:gcnt], val DOUBLE DEFAULT 0.0) AS
  SELECT flux FROM LOFARsrc[*][*][*][200][11][‘I’];

CREATE ARRAY CrCorr30_200 (idx INT DIMENSION[-fcnt+1:1:gcnt], val DOUBLE DEFAULT 0.0);
INSERT INTO CrCorr SELECT SUM(F.flux * G.flux) FROM F, G, CrCorr30_200 AS C GROUP BY
  F[MAX(0, -C.idx) : MIN(fcnt, gcnt-C.idx)], G[MAX(0,  C.idx) : MIN(gcnt, fcnt+C.idx)];
```

4. Conclusions

SciQL is a novel approach to provide a declarative language framework to bridge the gap between the relation model prevalent in RDBMSs and the array model underlying most mathematical packages. It greatly simplifies expression of complex scientific algorithms, leaving optimisation and execution to a mature database kernel.

This paper shows how a real-world complex astronomical application, the LOFAR TKP pipeline, can be modelled and manipulated in a concise manner using SciQL. By exposing the properties of array data to the RDBMS software stack, i.e. optimisers and kernel routines, SciQL opens up many opportunities to mine for real-time transient and variability searches. A prototype implementation is well under way and the TKP pipeline will be exercised on the SciLens platform.[2]

Acknowledgments. The present work is partly funded by the European projects PlanetData (http://www.planet-data.eu/) and TELEIOS (http://www.earthobservatory.eu/).

References

Gray, J., et al. 2006, The Zones Algorithm for Finding Points-Near-a-Point or Cross-Matching Spatial Datasets, Tech. Rep. MSR-TR-2006-52, Microsoft Research
Kersten, M., et al. 2011, in AD’11 (New York, NY, USA: ACM), 12
Scheers, L. H. A. 2011, Ph.D. thesis, University of Amsterdam
Zhang, Y., et al. 2011, in IDEAS2011 (New York, NY, USA: ACM), 10

[2] http://www.scilens.org/content/platform