Volume Title
ASP Conference Series, Vol. **Volume Number**
Author

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

Ying Zhang,[1] Bart Scheers,[1,2] and Martin Kersten[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. Data-intensive scientific research, such as in astronomy, calls for functional enhancements to DBMS technologies. In this paper we introduce SciQL (pronounced as 'cycle'), 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 in scientific domains, because it includes functionality often only found in mathematics software packages.

We demonstrate SciQL using examples taken from a real-life astronomical data processing system, e.g. the Transient Key Project (TKP) of the LOFAR radio telescope. In particular, we show how a a full Stokes spectral light-curve database of all detected sources can be constructed, by cross-correlation over multiple catalogues can be constructed. By exposing the properties of array data to the relational DBMS, SciQL also opens up many opportunities to enhance the data mining possibilities for real-time transient and variability searches.

1. Introduction

The ever growing use of high precision experimental instruments in astronomical projects, such as SDSS, LSST, Pan-STARRS and LOFAR, amounts to an avalanche of data to be stored, curated and analysed. Ingestion of gigabytes and even terabytes of data on a daily basis is taking place in many projects, while 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. Relational Database Management Systems (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 (see 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 generalization of window-based query processing with wide applicability in science domains, such as FT, PCA, 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(`www.lofar.org/astronomy/transients-ksp/transients-key-science-project`). LOFAR has two types of antennas: the Low Band Antennas (LBAs) operate at the frequency band 30 - 80 MHz, while the High Band Antennas (HBAs) operate at 120 - 240 MHz. The TKP focuses on studying the explosive and dynamic universe by observing transient and variable radio sources. One of the main goals of TKP is building a full Stokes spectral light-curve database of all detected sources, and therefore cross-correlating over multiple catalogues. Tens of gigabytes of extracted data needs to be stored in the light-curve database (Scheers 2011). For traditional RDBMSs, the array data oriented operations needed 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 Using SciQL Arrays

In this section, we show how the SciQL arrays are used to store the TKP data. Information derivation is discussed in Section 3.

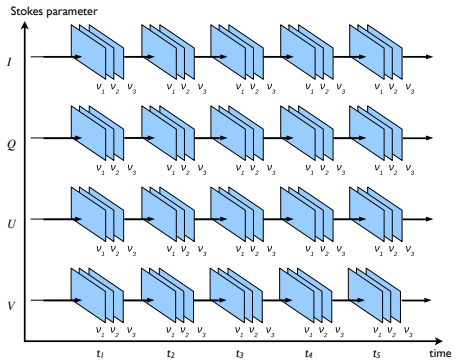

Figure 1. Schematic view of the TKP pipeline input streams of image cubes produces by one observation.

Catalogued Sources Every LOFAR observation produces images of a certain sphere area at different frequency bands over a time period for all four Stokes parameters, as depicted in Figure 1 The dataset at each Stokes parameter can be seen as streams of image cubes arriving at subsequent timestamps. Each image cube has the same observational timestamp, whereas the individual image planes in this cube fall in different frequency (sub)bands. In the LOFAR pipeline, source extraction and association algorithms are used 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, and it is stored in the array `LOFARsrc`:

```
CREATE ARRAY LOFARsrc(
  ra   DOUBLE DIMENSION[0:*:360],     decl DOUBLE DIMENSION[-90:*:90],
  ts   TIMESTAMP DIMENSION,           freq INT    DIMENSION[30:10:250],
  stks CHAR(1)   DIMENSION['I':*:'V'],
  CHECK (stks = 'I' OR stks = 'Q' OR stks = 'U' OR stks = 'V'),
  id INT, ra_err DOUBLE, decl_err DOUBLE, flux DOUBLE, ...);
```

The polar coordinates of the sources are denoted by the *dimensional attributes* (for short: *dimensions*) `ra` and `decl`, which meet the need that astrologists often search for

sources by their coordinates or in a certain area. The timestamps and frequencies of the measurements are respectively denoted by the `ts` and `freq` dimensions. A `CHAR` typed dimension `stks` is used to distinguish source values measured at different Stokes parameter. The `CHECK` constraint ensures that `stks` only contains the four Stokes parameters I, Q, U, V. All other measured properties of the sources are stored as *non-dimensional attributes* (for short: *attributes*), e.g., the `id` of each unique source, the errors of the polar coordinates (`ra_err, decl_err`), and the `flux`.

All sources from external catalogues are stored in the array `ExtCatSrc` below, which only differs from the array `LOFARsrc` with two more columns. The dimension `catname` distinguishes sources from different catalogues. The attribute `orig_catid` is the ID of a source taken from its originating catalogue, while the `id` attribute is a newly computed ID for this source, which is unique in this array. Currently, sources from VLSS, WENSS and NVSS are stored in this array. 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(
  ra   DOUBLE DIMENSION[0:*:360],      decl    DOUBLE DIMENSION[-90:*:90],
  ts   TIMESTAMP DIMENSION,            freq    INT         DIMENSION[30:10:250],
  stks CHAR      DIMENSION['I':*:'V'], catname VARCHAR(10) DIMENSION,
  CHECK (stks = 'I' OR stks = 'Q' OR stks = 'U' OR stks = 'V'),
  CHECK (catname = 'NVSS' OR catname = 'VLSS' OR catname = 'WENSS'),
  id INT DEFAULT 0, orig_catid INT DEFAULT 0,
  ra_err DOUBLE DEFAULT 0, decl_err DOUBLE DEFAULT 0, flux DOUBLE DEFAULT 0, ...);
```

Associated Sources An important operation in the TKP pipeline is to cross-correlate a LOFAR source with known sources in the major external catalogues, currently including the three aforementioned 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_id INT DIMENSION[0:1:*], vlss_id INT DIMENSION[0:1:*],
  wenss_id INT DIMENSION[0:1:*], nvss_id INT DIMENSION[0:1:*],
  w_dist DOUBLE DEFAULT NULL, s_idx DOUBLE DEFAULT NULL, ...);
```

In `AssocSrc`, we define the source IDs of each catalogue as a dimension to allow sources in one catalogue to be associated with sources from any number of other catalogues. The source IDs in all catalogues start from 1, while the values of all dimensions start from 0. A dimension value 0 means that the corresponding catalogue is excluded when computing the auxiliary values at a particular cell, which usage will be shown in the example below. All auxiliary values of an association is stored as non-dimensional attributes, e.g., the weighted dimensionless distance `w_dist` and the spectral index `s_idx`. An empty cell, i.e., all its non-dimensional attributes are `NULL`, means that it is unknown yet if the sources identified by this cell can be associated. A `w_dist` of -1 explicitly indicates that the sources identified by the cell's dimensions cannot be associated.

Assume the following associations:

LOFAR	VLSS	WENSS	NVSS
11	89	-	21

, i.e., the LOFAR source 11 is associated with the VLSS source 89 and NVSS source 21, but none of the WENSS sources. The results of the TKP source association algorithm are transitive, i.e., since the LOFAR source 11 is associated with the VLSS source 89 *and* the NVSS source 21, the VLSS source 89 is also associated with the NVSS source 21. The array `AssocSrc` (for short: `AS`)is designed in such a way that it can store the association and its auxiliary values of sources from *any* number of catalogues. The example here

contains four associations in total, whose auxiliary values are stored in the array cells `AS[11][89][0][0]`, `AS[11][0][0][21]`, `AS[0][89][0][21]` and `AS[11][89][0][21]`, respectively. Note that, since no association is found in WENSS, all cells `AS[11][89][*][21]`, except `AS[11][89][0][21]`, have `w_dist` = 1.

3. Multi-Catalogues Cross-Correlation

The TKP pipeline matches sources using three association parameters. For simplicity, we only use the weighted dimensionless distance here, defined as: $r = \sqrt{\frac{(\Delta\alpha)^2}{\sigma^2_{\Delta\alpha}} + \frac{(\Delta\delta)^2}{\sigma^2_{\Delta\delta}}}$ 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 (but no associations among external catalogues).

```
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)*COS(AVG(L.decl)) - E.decl*COS(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.ra-@ra_delta:L.ra+@ra_delta][L.decl-@dc_delta:L.decl+@dc_delta], 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`. This reduces the dimensions in the resulting arrays. All omitted attributes are selected. Then, for every LOFAR source, a group is constructed with every nearby external sources, using an array tiling in the `GROUP BY` clause. A credible association is immediately inserted into `AssocSrc` together with the auxiliary information `w_dist` and `s_idx`. If a qualified external source is from one catalogue, its `orig_id` in the other catalogues are 0. Thus, in the `SELECT` clause, only one of the `orig_id`s can be non-zero.

4. 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 following query builds a spectrum of Stokes I of the LOFAR source 11. The frequencies of the external catalogues are added as constants.

```
SELECT * FROM (
  SELECT freq, AVG(flux) AS flux FROM AssocSrc[11] AS A, LOFARsrc[*][*][*][*]['I'] AS L
  WHERE L.id = 11 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_id
     OR E[*][*][325]['WENSS'].orig_id = A.wenss_id
     OR E[*][*][1400]['NVSS'].orig_id = A.nvss_id
) AS spectrum
ORDER BY freq;
```

To retrieve data for a light-curve graph, only the LOFAR catalogue can be used. However, 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 source association example in Section 2, in which the LOFAR source 11 is associated with the NVSS source 21. We are interested in the similarity of the flux of the NVSS source 21 over time in the frequency bands 30 MHz (LOFAR low band) and 200 MHz (LOFAR high band). The following query computes the cross-correlation of the two time-series at these frequency bands:

```
DECLARE f_max INT, f_cnt INT, g_cnt INT;
SET f_max = SELECT MAX(ts) FROM LOFARsrc[*][*][*][30]['I'] WHERE id = 11;
SET f_cnt = SELECT COUNT(*) FROM LOFARsrc[*][*][*][30]['I'] WHERE id = 11;
SET g_cnt = SELECT COUNT(*) FROM LOFARsrc[*][*][*][200]['I'] WHERE id = 11;

CREATE ARRAY VIEW F (idx INT DIMENSION[0:1:f_cnt], flux DOUBLE DEFAULT 0.0) AS
  SELECT flux FROM LOFARsrc[*][*][*][30]['I'] WHERE id = 11;
CREATE ARRAY VIEW G (idx INT DIMENSION[0:1:g_cnt], val DOUBLE DEFAULT 0.0) AS
  SELECT flux FROM LOFARsrc[*][*][*][200]['I'] WHERE id = 11;

CREATE ARRAY CrCorr30_200 (idx INT DIMENSION[-f_max:1:g_cnt], 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(f_cnt, g_cnt-C.idx)],
                            G[MAX(0,  C.idx) : MIN(g_cnt, f_cnt+C.idx)];
```

5. 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 optimization and execution to a mature database kernel.

In this paper, we show how a real-world complex astronomical application, the LOFAR TKP pipeline, can be modelled and manipulated in a concise manner using SciQL. Exposing the properties of array data to the RDBMS software stack, i.e. optimizers 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 (`http://www.scilens.org/content/platform`).

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

References

Kersten, M., Zhang, Y., Ivanova, M., & Nes, N. 2011, in Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases (New York, NY, USA: ACM), AD '11, 12

Scheers, L. 2011, Ph.D. thesis, University of Amsterdam

Zhang, Y., Kersten, M., Ivanova, M., & Nes, N. 2011, in International Database Engineering & Applications Symposium (IDEAS2011) (New York, NY, USA: ACM), 10