Astronomical Data Analysis Software and Systems XV **P.144**
ASP Conference Series, Vol. XXX, 2005
C. Gabriel, C. Arviset, D. Ponz and E. Solano, eds.

Q3C, Quad Tree Cube – The new sky-indexing concept for huge astronomical catalogues and its realization for main astronomical queries (cone search and Xmatch) in open source database PostgreSQL

Sergey Koposov[1], Oleg Bartunov

Sternberg Astronomical Institute, Universitetskiy pr. 13, 111992, Moscow, Russia

Abstract. In this paper we present Q3C (Quad Tree Cube) – the new sky indexing scheme and its implementation for open source database PostgreSQL. We have implemented our sky-partitioning scheme as loadable module for PostgreSQL and developed the very simple SQL interface for main astronomical queries: cone search, various spatial searches on the sphere and cross-matches of catalogues. The performance of our realization is very high and allow to work easily with the largest existing catalogues (USNO-A/B, 2MASS, SDSS). We propose PostgreSQL and our sky-indexing scheme as open source solution for any huge databases, VO-services dealing with the huge catalogues, SkyNodes.

1. Introduction

In last few years, several projects have provided to community a set of very large catalogues consisting from up to billion of objects. Since then, the simple access methods are not more suitable to work with that amount of data. The databases are needed to store it. But even the most advanced databases are lacking for the methods to work effectively with spherical astronomical data and astronomical queries, like cone searches and cross-matches. So special astronomical indexes and sky-partitioning schemes were invented for that purpose. The first successful attempt was HTM – hierarchical triangular mesh (Kunszt 2000). That sky partitioning scheme has been used in MS SQL to provide the interfaces for main astronomical queries in SDSS project. But in fact, HTM has several disadvantages (Mullane 2000): first, HTM is too complicated scheme and the computations significantly limit the performance of it (especially on high depth of segmentation); second, HTM was specifically developed for Microsoft SQL server, which is not open source and nor free. That was the motivation for the project of implementing a new simple, fast and powerful sky-indexing scheme for open source database PostgreSQL.

[1]Max Planck Institute for Astronomy, Königstuhl 17, D-69117, Heidelberg, Germany

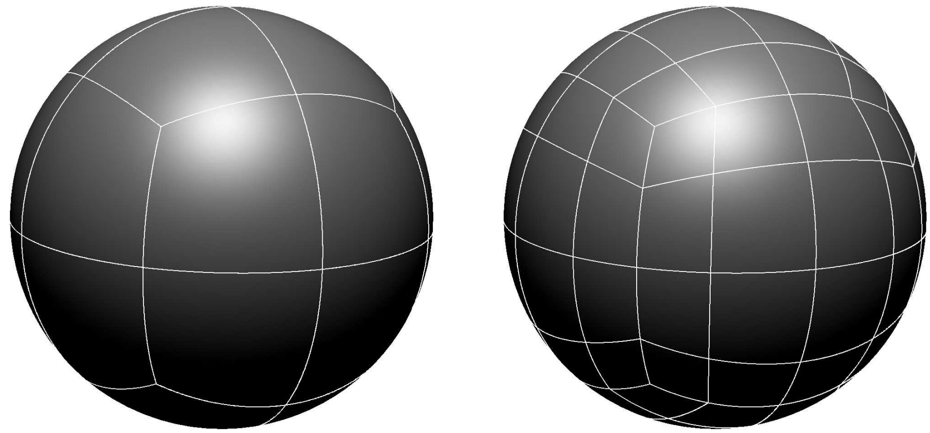

Figure 1. The sphere segmentation in Q3C

2. Pixelization Scheme Details

The idea of it is similar to other sky-indexing/sky pixelization schemes. The base of the scheme is the cube inscribed in the sphere. And on each face of the cube the quad tree is constructed. The quad tree structure create the mapping of the 2D coordinates in the square to the bitmask (or just integer number). Since there are 6 faces, the 3 bits indicating the face number are appended. Thus the mapping of the cube to the integer numbers is established. And since one can easily do the central projection of the surface of the cube to the sphere, automatically that quad tree structure is inherited by the sphere. The final sky pixelization of the sphere with different depths is shown in the figure 1. We want to emphasize several important points of this scheme. The scheme is extremely simple, and all the mathematic calculations are simple too (the quad tree in the square is very simple to work with, and the trigonometric operations are not too numerous since the mapping of the sphere and cube surface is just the central projection). We emphasize that the computations are much simpler than in HTM and HEALPIX(Górski 2005) and thus the computations do not limit the performance of the queries. Due to the usage of the quad tree in the square the special look-up tables speeding up the computations are utilized, and that's why our scheme work certainly faster in the case of high depth of segmentation (e.g. as compared with HTM). The individual pixels of our pixelization scheme do not have equal areas (as HTM), but the property of equal areas of the pixels is absolutely unnecessary when the scheme is used for database indexing).

As we have seen, Q3C provide the mapping of each point of sphere to the integer number (we call it IPIX value)(with such a property that nearby points on the sphere have nearby IPIX values. And that value can be used to create the index to allow the fast searches on the sphere. To effectively use that index every spatial query is is first segmented on pixels of the segmentation (the example for the circular query is on figure 2), and since each pixel represent the continuous range of IPIX, the data for that part of the sphere is easily and quickly retrieved from the database using the index.

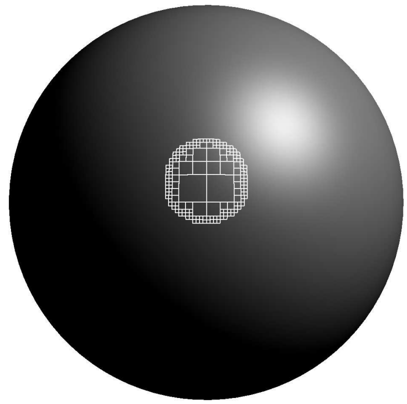

Figure 2. The example showing how the cone query is handled using Q3C

3. Realization of Q3C in PostgreSQL and Main Spatial Queries

The sky-indexing scheme Q3C has been programmed in C for the maximum performance. The special functions integrating Q3C in PostgreSQL have been written. Now Q3C is the plugin for PostgreSQL which can be simply deployed for the database. In the Q3C package we have written the set of SQL functions for the main spatial queries: the cone search, rectangle query, cross-matches and others. We emphasize that in the database level Q3C do NOT require addition of special columns to the database, and only one index should be created! The example here show how easy is to deploy Q3C for the PostgreSQL databases/tables (assuming that you have the table usnob with the ra, dec columns)

```
db# CREATE INDEX usnob_idx ON usnob (q3c_ang2ipix(ra,dec));
db# CLUSTER usnob_idx on usnob;
db# ANALYZE usnob;
```

Now all the special queries from Q3C can be used. For example to perform the cone search around the point (10 deg, 30 deg) with 1deg radius:

```
db# SELECT * FROM usnob WHERE q3c_circle_query(ra,dec,10,30,1);
```

Or, to perform the cross-match of usnob with other table 2mass with error radius of 1 arcsec (=0.00027deg):

```
db# SELECT * from 2mass, usnob WHERE
q3c_join(2mass.ra, 2mass.dec, usnob.ra, usnob.dec, 0.00027);
```

With the same syntax you can do OUTER join cross-matches, or the cross-matches with variable error radius. Q3C also support the rectangular queries, and will support polygonal area queries soon.

4. The Performance of Q3C with PostgreSQL

Our main point that Q3C is not only the first open source sky-indexing scheme but it is extremely fast. This main factors producing such performance are: the ordering of the data induced by Q3C guarantee the optimal I/O performance of the data retrieval from the database; due to the simplicity of the sky-partitioning

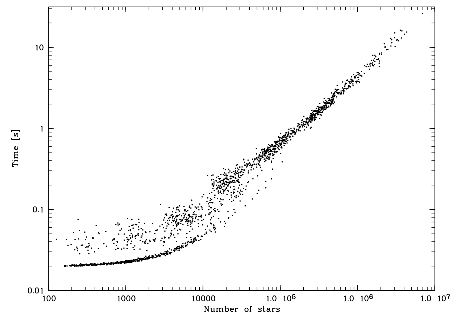

Figure 3. The performance of cone search and cross-match queries. On the left panel, the time to perform the cone search query versus the number of stars returned is shown. On the right panel, the time to perform the 1 arcsecond cross-matches between the catalogues of different sizes with USNO-B versus the number of stars in the catalogue.

scheme, the computations (CPU time) does not limit the performance of Q3C even on high depth of segmentation; PostgreSQL works very effectively with the complex queries produced by Q3C. The figure 3 shows the performance of Q3C for the main astronomical queries (cross-match and cone-search).

5. Conclusions

- Q3C with PostgreSQL are the first open source solutions for large databases, allowing to perform the spatial queries on the sphere for large databases.
- Q3C provide the very simple interface to the main astronomical spatial queries
- To use Q3C the schemas of tables need not to be altered. Only one B-tree index should be created
- Q3C is very fast and works perfectly even with the largest catalogues.
- The depth of the segmentation is limited by 30 (but in principal it can be easily extended). Thus the size of the smallest pixel in the segmentation is $\approx 1mas \times 1mas$).
- ADQL to SQL translation for cone searches and Xmatches is very straightforward and it is very easy to build the Skynode based on Q3C.
- Q3C can be downloaded from `http://q3c.sourceforge.net`

This work was partially supported by Russian Foundation for Basic Research, grant 05-07-90225-в.

References

P. Z. Kunszt et al., 2000, in ASP Conf. Ser., Vol. 216, ADASS IX, ed. N. Manset, C. Veillet, & D. Crabtree (San Francisco: ASP)

W.O'Mullane et al., 20 MPA/ESO/MPE Joint Astronomy Conference, Garching 2000

K.M. Górski, Eric Hivon, A.J. Banday, B.D. Wandelt, F.K. Hansen, M. Reinecke, M. Bartelmann, 2005, ApJ, 622, 759