PostgreSQL: the Suitable DBMS Solution for Astronomy and Astrophysics

Chilingarian, I.[1) 2) +]; Bartunov, O.[1) 3) *]; Richter, J.[4]; Sigaev, T.[3) *]

[1)] Sternberg Astronomical Institute of the Moscow State University;[2)] Special Astrophysical Observatory of the Russian Academy of Sciences;[3)] Delta-Soft LLC;
[4)] International Meteor Organization;[+)] A participant of MIGALE collaboration, responsible for DBMS subsystem;[*)] An official member of PostgreSQL development team

Abstract:

PostgreSQL, the open-source ORDBMS, is probably one of the best database solutions for astronomy and astrophysics. Compared to several available commercial and non-commercial database engines, it appears to be the most versatile.

At present PostgreSQL is being used in several well-known astronomical projects, for example in the HyperLEDA database, http://leda.univ-lyon1.fr/ and in the MAPS, http://aps.umn.edu/ (Minnesota Automated Plate Scanner) Catalog of the POSS1.

Extensibility is the most remarkable feature of this RDBMS - it allows to develop custom data types, queries and indexed access methods, optimized for specific tasks, without knowledge of database internals. This is very important for "non-standard" tasks, typical for scientific reaserch.

We present two backend programmed modules.

- pgSphere, the ready-for-use contribution module for PostgreSQL, offers the capability for dealing with geometric objects in spherical coordinates. This module demonstrates all the possibilities of backend programming using the GiST interface for spatial indexing of spherical data. PgSphere can be very useful in astronomy and the geo-sciences. Performance tests are included.
- pgAstro, the contribution module based on the pgSphere engine, provides astronomy-specific functions and methods, for example positional astronomy and physical models.

We show that PostgreSQL is the most advanced database solution for astronomical applications and would be very useful for different Virtual Observatory projects.

1. What is PostgreSQL and why we are using it?

PostgreSQL is an object-relational database management system (ORDBMS) based on POSTGRES, Version 4.2, developed at the University of California at Berkeley Computer Science Department. PostgreSQL provides SQL92/SQL95 language support and numerous powerful features making it well-suited for different scientific and technological tasks. A lot of object-relational concepts assisting in modern commercial databases were pioneered in POSTGRES.

The main concepts and features of PostgreSQL:

- rapidly developing open source freely distributed DBMS
- object-relational concepts
- SQL support
- simple and handy front-end interfaces for different software platforms (C, Java, Perl, Python, ODBC)
- extensibility of the DB server functions, i.e. developing of custom data types and data access methods

These features allow PostgreSQL to be used in different scientific projects. At present it is being used in the following projects related to astronomy and astrophysics:

- ✓HyperLEDA database, http://leda.univ-lyon1.fr/ , a part of MIGALE project. PostgreSQL is used to store LEDA and Hypercat catalogue data for more than 3 millions of objects and to provide an access to these data. One of us is responsible for the DBMS subsystem development in HyperLEDA.
- ✓SAI Astronomical Databases, http://www.sai.msu.su/database.html The idea was to provide an online access to the databases developed in Sternberg Astronomical Institute (Moscow, Russia).
- ✓TASS, The Amateur Sky Survey, http://www.tass-survey.org/ PostgreSQL is used to store and access the data for all objects found on the images every night. PgSphere extention is used in this project.
- ✓MAPS, Minessota Automated Plate Scanner, http://aps.umn.edu/ PostgreSQL is used to store the catalogue of the objects from POSS1 plates. Special data types and access methods were implemented for effective data access.

During last years a huge amount of observational astronomical data appeared as results of surveys. So the creation of effective access methods for the heterogeneous data has become a very important task for CS specialists. It is desirable to have some standard abstraction layer for such methods and SQL approach seems to be quite versatile for this kind of problems.

Usually scientific objectives imply the work with data types different from integer and floating point numbers, strings, timestamps and money provided by the standard SQL. For example, many of astronomical and astrophysical tasks require effective work with celestial coordinates. This implies the 2-dimensional indexing of the positions on sphere to achieve high performance on large datasets. Unfortunately, there is no standard solution for this problem in the modern DBMSs and there is no standard data types even for 2-dimensional objects on cartesian plane. So, the extensibility of the DBMS becomes the most valuable feature. Let's consider several database solutions available on the market to compare them and conclude about their suitability for astronomy.

A good review on this problem was made by Clive Page [1]. Taking it into the account it is possible to make the following conclusions:

- Oracle is a market leader. It is full featured database solution, it is extensible and high-performance, it has support for user-written data types and access methods and for GiST access methods, but it is very expensive.
- MS SQL Server has similar feature set, but it can be used for Windows only. SDSS project successfully uses this DBMS. It is also quite expensive.
- DB2 (and Informix). It is another example of full featured solution, and it is used in several large astronomical projects, such as NED, http://nedwww.ipac.caltech.edu/ Again, it is very expensive.
- Sybase is similar to DB2 by the abilities, it is traditionally used in many astronomical applications.
- MySQL is open source RDMBS with a reputation for efficiency. It uses BerkeleyDB in the lower layer. But is not extensible and feature set is rather poor. Anyway it is quite efficient and cheap solution for static datasets.
- PostgreSQL is open source, has extremely rich feature set, it is easily extensible, supports GiST access methods. PostgreSQL is easy to install and configure. Unfortunately, many of its features are poorly documented.

So, PostgreSQL is the only extensible free open source DBMS solution. This opens the unique opportunities in the data distribution. For instance, PostgreSQL is supplied as an additional package with Pleinpot software used to provide an access to HyperLEDA data. So, end-user gets the configured database engine together with the data. Thanks to this, Pleinpot with HyperLEDA is now the only product in the market providing fast access to homogenized data on more than one million of galaxies and name based cross-identification tools for them among ~70 catalogues.

4. Concepts of pgAstro

Using pgSphere module it becomes possible to solve some astronomical tasks using SQL queries.

We are introducing pgAstro contribution module, distributed under GPL2 license. It will be a set of tools on SQL-layer and backend layer devoted to astronomical tasks. Two possible applications are clear now:

- ➤Positional astronomy. Some astrometric functionality will be included, for instance, it will be possible to do cone search for a given epoch and equinox taking into the account proper motions of the stars, to calculate precession and nutation on the fly, to check if the given object belongs to the given constellation etc.
- ➤Coordinate based cross-correlation [4]. This task is very important for identifying objects in different catalogues. The idea is very simple: to cross-identify the list of objects with a given catalogue one should check for each object in the list if it belongs to any of the circles with radii equal to r_{corr} centered on the objects from the catalogue.

We are looking for other astronomical applications for pgSphere.

References:

1. Clive Page. "Indexing the Sky", http://www.star.le.ac.uk/~cgp/ag/skyindex.html
2. A. Baruffolo an L. Benacchio, "Object-Relational DBMSs for Large Astronomical Catalogue Management", Proceedings of the ADASS-VII, ASP Conference Series, Vol. 145, 1998
3. A. Baruffolo, “R-Trees for Astronomical Data Indexing”, Proceedings of the ADASS-VIII, ASP Conference Series, Vol. 172, 1999
4. Clive Page, “A New Way of Joining Source Catalogs using Relational Database Management System”, Proceedings of the ADASS-XII, ASP Conference Series, Vol. 295, 2002
5. Walid G. Aref at al., “SP-GiST. A General Index Framework for Space Partitioning Trees”, http://www.cs.purdue.edu/homes/aref/dbsystems_files/SP-GiST/

2. Extensibility of PostgreSQL

As noted before, the extensibility becomes the most important feature of the DBMS to be used in science. PostgreSQL provides very wide possibilities for extending the database and adopting it to the raised objective.

- ✓PostgreSQL allows to create user-defined functions and aggregates in the upper layer using SQL or one of the available procedure languages. This feature is quite common for the most of the DBMSs. Also it is possible to create custom data types and use these high level functions for dealing with them.
- ✓PostgreSQL provides a powerful functionality for so called back-end programming. This allows developer to create functions using low-level language (i.e. C), compile them and load dynamically into the running database server as shared objects. Binary code usage increases the performance dramatically. Moreover, the standard interface to GiST (Generalized Search Tree) is provided to create custom data types with indexed access methods and extensible set of queries for specific domain experts not a database one.

GiST was implemented in an early version of PostgreSQL by J. Hellerstein and P.Aoki, more details is available from "The GiST Indexing Project" (http://gist.cs.berkeley.edu/) at Berkeley.

As an "university" project it had a limited number of features and was in rare use. Since version 7.1 of PostgreSQL the GiST support was taken up by Oleg Bartunov and Teodor Sigaev. Current implementation of GiST supports:

- Variable length keys
- Composite keys (multi-key)
- It provides NULL-safe interface to GiST core

But GiST cannot be used to implement such well known multi-dimensional indexing methods as HTM (Hierarchical Triangular Mesh), because HTM is a kind of Space Partitioning Trees. More general index structure called SP-GiST exists for dealing with SP-Tree algorithms [5]. It also can be implemented as extension to PostgreSQL. We have discussed this with people from CS department of Purdue University-West Lafayette, Indiana. They are interested in making this development.

Several extensions to PostgreSQL based on GiST interface exist. They are described here: http://www.sai.msu.su/~megera/postgres/gist/

We'll emphasize the pgSphere extension, useful for astronomy more then the others.

3. PgSphere project

We have developed pgSphere contribution module, http://www.pgastro.org/cgi-bin/wiki.pl?pgSphere for PostgreSQL using backend programming and GiST interface. It is distributed under BSD license. It introduces data types for geometrical objects on a sphere and access methods for them.

The project is hosted by Gborg, http://gborg.postgresql.org/

pgSphere provides the following functionality:

- input and output of "spherical" data in several formats (radians, degrees etc.)
- containing, overlapping and other geometrical operations for different types of objects on a sphere; some of these types are shown below
- various input and converting functions and operators
- calculation of circumference and area of an object on a sphere
- spherical transformations
- indexed data access methods for "spherical" data types

Hence it is possible to do a fast search and analysis for objects with spherical attributes, using PostgreSQL. This functionality may be very useful for different types of astronomical and geo-science applications. For instance it makes possible management of data for geographical objects on the Earth or astronomical data like stellar and other catalogues conveniently using a SQL interface.

The aim of pgSphere is to provide a uniformed access to spherical data. PostgreSQL itself supports a lot of software interfaces; therefore one now can use the same database for access with different utilities and applications.

Several performance tests were made with different datasets. We used TYCHO-1 catalogue and its parts to compare the performance of GiST R-tree based algorithm implemented in pgSphere to 2-column B-tree index on celestial coordinates. The whole catalogue includes 1055115 stars. Four sub-catalogues were created using the simple schema:

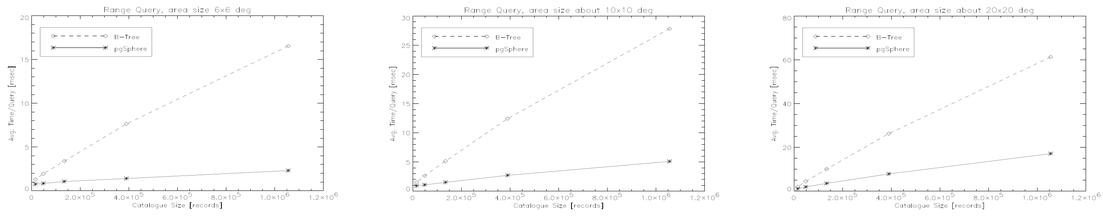
```
SELECT * INTO TABLE tycho10 FROM tycho WHERE mag<10;
SELECT * INTO TABLE tycho09 FROM tycho WHERE mag<09;
SELECT * INTO TABLE tycho08 FROM tycho WHERE mag<08;
SELECT * INTO TABLE tycho07 FROM tycho WHERE mag<07;
```

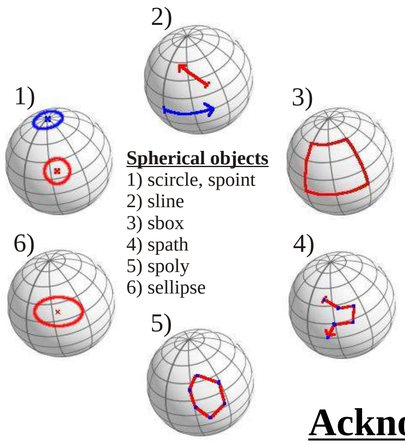
The number of objects in the sub-samples varies from ~15000 to ~400000. After construction of indices these two queries had been executed for the whole catalogue and all the sub-samples for 10 times:

```
SELECT count(ra) FROM tycho WHERE pos @ sbox '((183d,3d),(177d,-3d))';
SELECT count(ra) FROM tycho WHERE (ra BETWEEN 177 AND 183) AND (dec BETWEEN -3 AND 3);
```

The execution times were averaged for each group of queries. Three series of queries were executed for the area on the sphere of 6x6, 10x10 and 20x20 degrees. The results are demonstrated below. This graph has exactly the same meaning as fig.1 in [2]. After 5 years the absolute running time decreased by a factor of 500 due to progress in hardware and software, but the relative performance ($t_{B\text{-}tree}/t_{R\text{-}tree}$) is more or less the same.

PgSphere is close to the first stable release now, and we hope to make it before January 2004.





5. Conclusions

From the given examples PostgreSQL appears to be the most versatile DBMS solution for astronomy and astrophysics. It is easily extensible, has powerful set of features well comparable to leading commercial database solutions. The fact that PostgreSQL is freely distributed open source software indicates a very important advantage. Many people can create contributions useful for scientists, which is hardly possible with any commercial databases.

The further features of PostgreSQL will include XML support. It may be very useful for many VO applications and tools.

Acknowledgments

Our development is supported by the Russian Foundation for Basic Research, projects #02-07-90222 and #03-07-06116. Also we greatly appreciate PostgreSQL community, TASS Amateur Sky Survey working group, especially Robert Creager and Chris Albertson. Great thanks to ADASS-XIII organizing committee for financial support.

Any feedback is welcome at chil@sai.msu.ru