

Object–Relational DBMSs for Large Astronomical Catalogue Management

A. Baruffolo and L. Benacchio

Astronomical Observatory of Padova, Italy

Abstract. Astronomical catalogues containing from a million up to hundreds of millions of records are becoming commonplace. While they are of fundamental importance to support operations of current and future large telescopes and space missions, they appear also as powerful research tools for galactic and extragalactic astronomy. Since even larger catalogues will be released in a few years, researchers are faced with the problem of accessing these databases in a general but efficient manner, in order to be able to fully exploit their scientific content. Traditional database technologies (i.e. relational DBMSs) have proved to be inadequate for this task. Other approaches, based on new access technologies, must thus be explored. In this paper we describe the results of our pilot project aimed at assessing the feasibility of employing Object–Relational DBMSs for the management of large astronomical catalogues.

1. Introduction

Large astronomical catalogues, with one million up to hundreds of millions of records, are becoming commonplace (e.g., Tycho, GSC I, USNO–1.A). They have an obvious operational use, in that they will be employed throughout the cycle of observations of the next generation large telescopes and space missions for proposal and observation preparation, telescope scheduling, selection of guide stars. However, they appear also as powerful research tools for the study of the local and grand scale structure of the Galaxy, cross–identification of sources etc.

Since even larger catalogues will be available in the near future (e.g., GSC II), we are faced with the problem of accessing these databases in an efficient but general manner. If the scientific content of these catalogues is to be fully exploited, astronomers must be allowed to issue almost any query on the database without being hampered by excessively long execution times.

2. The Large Astronomical Catalogues Management Problem

There are many possible approaches to the problem of managing very large astronomical catalogues.

Data can be organized in a catalogue specific file structure and accessed by means of programs. This approach allows for fast access for a defined set of queries, on the other side new queries require writing of programs and access is limited to one catalogue only.

One can then consider the use of “custom”, astronomical, DBMSs, like, e.g., DIRA (Benacchio 1992) or Starbase (Roll 1996). They support astronomical data and queries, and are freely available. However they typically do not support large DBs, since data is often stored in flat ASCII files and secondary access methods are usually not provided.

Commercial Relational DBMSs have also been used in the past: they are robust systems, widely used in the industry, whose data model is close to the structure of astronomical catalogues (Page & Davenhall 1993). On the other side, they have limited data modeling capabilities, and their access methods support indexing on simple data types and predefined set of query predicates. Their use with large astronomical catalogues has proved to be problematic (Pirenne & Ochsenbein 1991).

Another possible approach is to use an Object–Oriented DBMS. These systems feature a powerful data model, which allows data and operations to be modelled. However, they do not provide an efficient query processing engine, such facility must be implemented on the top of the DBMS (Brunner et al. 1994).

3. An Object–Relational Approach

From the discussion above it is apparent that, in order to give astronomers a general and efficient access to new databases, a DBMS must be employed that support astronomical data and queries, and that is able to efficiently execute them.

Recently, a new class of DBMSs has emerged, Object–Relational DBMSs, that provide:

- user–defined data types and functions which allow to define methods to create, manipulate and access new data types;
- user–defined index structures that can speed up the execution of queries with predicates that are *natural* to the new data types;
- an extensible optimizer that can determine the most efficient way to execute user queries.

In a word, they provide powerful data modeling and efficient query processing capabilities. We thus conducted a pilot project aimed at assessing the feasibility of employing ORDBMSs in managing large astronomical catalogues.

We built a prototype catalogue management system on a Sun Sparc Ultra 1/140, equipped with 128 MB RAM and 10 GB HD, using PostgreSQL 6.0 (Yu & Chen 1995), as the Object–Relational DBMS. Software was developed, in the C language, for the custom data types and functions, to extend the DB B–tree index to support astronomical coordinates, and to implement a two dimensional R–tree (Guttman 1984) index on coordinates on top of the DB GiST (Hellerstein et al. 1995) secondary access methods.

We defined in the DBMS typical astronomical data types (e.g., coordinates) and implemented functions acting on them. The DB query language was then extended by bounding these functions to user–defined operators so that they could be employed in formulating queries. For example, typical astronomical queries that were supported in this way are:

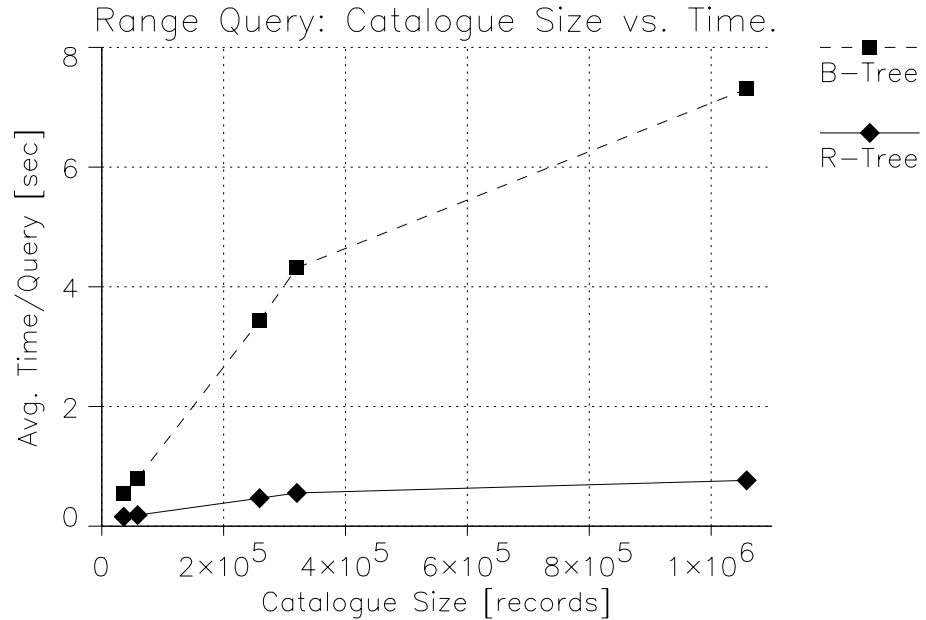


Figure 1. Execution times for range queries, covering an area of $40^{\circ 2}$ in the sky, over coordinates on catalogues of various sizes indexed using B-Trees (dashed line) and R-Trees (solid line). Times were averaged over 100 queries and include actual retrieval of the data from disk.

- Range query on coordinates:

```
SELECT * FROM AGK3 WHERE POS -> '(0:4,-55:16,0:8,-54:16)';
```
- Search-by-cone:

```
SELECT * FROM AGK3 WHERE POS @> '(0:4,-55:16,0.1)';
```
- Cross-correlation of catalogues based on sky position:

```
SELECT A.ID,B.ID,A.POS,B.POS FROM AGK3 A, TYCHO B WHERE
A.POS @> SkyPosToSkyCone(B.POS,0.1);
```

In order to evaluate the performance improvement that can be obtained by employing multi-dimensional indexes, we created and populated a database with data from five catalogues, ranging in size from ~ 35000 records (IRS) up to a million records (Tycho). From these catalogues we extracted: ID, α , δ , μ_α , μ_δ , σ_α , σ_δ , σ_{μ_α} , σ_{μ_δ} , magnitude and spectral type. All catalogues were then indexed on coordinates using both B-Trees and R-Trees, and a series of tests were run to measure the performance of these access methods.

Results for one of these tests are shown in Figure 1. From this graph it is apparent that, even for a simple range query over coordinates, execution times are greatly reduced when using an R-tree index with respect to a B-tree index, which is the access method commonly employed in relational DBMSs.

It is to be noted that absolute query execution times are only indicative of the DB performance, because they depend on the actual content of the catalogues, system hardware, etc. Relative performance of the R-tree based indexes with respect to B-trees is more significant, because all other conditions are identical. Another important point is that other typical astronomical queries besides the simple range query (e.g., search-by-cone) can take advantage from the presence of an R-tree based index, while their execution can't be speeded up using B-tree indexes.

4. Conclusions

Our experience in employing an ORDBMS to manage astronomical catalogues has been positive. The data modeling capabilities of this DBMS allow to define typical astronomical data in the DB. We verified that it is possible to extend the DB query language with astronomical functionalities and to formulate queries with astronomical predicates. Further, the execution of these queries is speeded up by the use of multidimensional indexes. Performance improvements, with respect to traditional access methods, are apparent even with small catalogues.

On the minus side, it should be noted that substantial effort is required to add new index structures to the DB, however some commercial ORDBMSs already supporting R-Trees and other third party "extensions" (with access methods) are also available. They can be customized to support astronomical data and predicates.

We also experienced long data loading and index building times, this is an architectural issue though, it is not a fundamental limitation due to the specific data model of the DBMS. In fact, commercial ORDBMS usually provide parallel operations for data loading, index creation and query execution.

The bottom line is that in our experience ORDBMS provide the basic building blocks for creating systems for an efficient and general access to large astronomical catalogues. We think that this technology should be seriously taken into account by those planning to build such systems.

References

- Benacchio, L. 1992, ESO Conf. and Workshop Proc. 43, 201
- Brunner, R. J., Ramaiyer, K., Szalay, A., Connolly, A. J., & Lupton, R. H. 1995, in ASP Conf. Ser., Vol. 77, Astronomical Data Analysis Software and Systems IV, ed. R. A. Shaw, H. E. Payne & J. J. E. Hayes (San Francisco: ASP), 169
- Guttman, A. 1984, Proc. ACM SIGMOD, 47
- Hellerstein, J. M., Naughton, J. F., & Pfeffer, A. 1995, Proc. Int. Conf. on VLDBs, 562
- Pirenne, B., & Ochsenbein, F. 1991, ST-ECF Newsletter, 15, 17
- Page, C. G., & Davenhall, A. C. 1993, in ASP Conf. Ser., Vol. 52, Astronomical Data Analysis Software and Systems II, ed. R. J. Hanisch, R. J. V. Brissenden & Jeannette Barnes (San Francisco: ASP), 77

Yu, A., & Chen, J. 1995, PostgreSQL User Manual

Roll, J. 1996, in ASP Conf. Ser., Vol. 101, Astronomical Data Analysis Software and Systems V, ed. George H. Jacoby & Jeannette Barnes (San Francisco: ASP), 536