Performance Comparison between Relational and Object-Oriented Databases

Vipin Saxena, PhD
Department of Computer Science
B.B. Ambedkar University (A Central University)
Rae Barely Road, Lucknow-25, U.P. (India)

Ajay Pratap
Department of Computer Science
B.B. Ambedkar University (A Central University)
Rae Barely Road, Lucknow-25, U.P. (India)

ABSTRACT

An Object-Oriented database can utilize the benefits of both the design and implementation of any application. Due to the increased popularity of database systems many new database systems based on varying data model and implementation have entered in the market. Database systems have complex architecture but they are the key factors behind the business transformations. Choosing the best one in any category is an important task based on performance analysis. This chapter deals with the database estimation methodology which integrates the database analysis task and performance analysis task. There are three major techniques for the performance estimation which are analytical modeling, simulation modeling and benchmarking.

General Terms

Performance Comparison, Response Time, Algorithms et. al

Keywords

DB4O, SQL Server, Writing Objects, Updating Objects, Retrieving Objects

1. INTRODUCTION

A database management system (DBMS) is a collection of logically related data and software to access and manipulate those data. Data modeling is a conceptual method of structuring the data by presenting the association between various entities and their attributes. It also consists of static properties, integrity rules and dynamic properties of applications. They are broadly classified into three categories which are as following:

- Record-based Data Model
- Object-based Data Model
- Physical Data Model

Overall logical structure of database is represented with the help of record-based data model which can explicitly specify the data integrity constraints. This data model contains different types of fixed format records and is classified in three categories which are as following:

- Hierarchal data model
- Network data model
- Relational data model

1.1 Object-based Data Model

Data and relationship among the data and its attributes is presented with the help of object-based data model. It is also capable to explicitly specify the data integrity constraints and has flexible data structure capabilities. It is divided into four categories but entity-relationship model and object-oriented model are the main techniques. All techniques of object-based data models are as following:

- Entity-Relationship Model
- Semantic Model
- Functional Model
- Object-Oriented Data Model

2. LITERATURE SURVEY

Performance evaluation of any database management system is a very critical task and in this section various related research papers are discussed. A methodology for evaluating the performance of database management systems in a multi-user environment has proposed by Boral and DeWitt [1].

The performance of an object database using commercial Ontologic's Vbase object database platform has been demonstrated by Duhl and Damon [2]. The benchmark results has been compared with the existing relational database systems, which show that object databases are capable of performing better than relational database systems. Critical aspects of object-oriented DBMS performance have been discussed by Cattell [3]. A benchmark has been designed and used for both relational and object-oriented databases. Problem related to the object identity and theoretical framework of fundamental of object oriented databases has been presented by Zdonik and Maier [4]. Ullman [5] has compared the deductive and object-oriented approaches to new database systems and contrasted the approaches with regard to the way they classify data elements.

Ishikawa [6] has described the concepts of object-oriented database system. Various architectural, design and implementation issues have been discussed. Chaudhri et al. [7] have redressed the suitability of ODBMS implementations for other application domains and evaluated two pure ODBMSs and one hybrid Object-Relational DBMS to determine the suitability. Gorla [8] has developed a methodology for the design of an efficient storage structure of object-oriented databases that minimized the cost of database operations. The genetic algorithm has been used to the methodology. The developed methodology has been applied on the University database and the outputs are compared with the previous storage model which shows around 26% to 31% improvement in the performance. A performance evaluation framework that allows a system designer to predict performance at several steps in the design process has been introduced. Smith and Williams [9] have presented a software performance engineering which is a systematic and quantitative approach for constructing software system that meet performance objective.

Bagui [10] have discussed about the object oriented database. Achievements and weaknesses of object-oriented database have been discussed in detail which helps one to decide the purpose of choosing right object-oriented database software for their purpose. Yin and Ray [11] have presented an extension of UML data modeling profile and explained the modeling of relational database operations using UML. A framework is presented to model atomic database operations, which are used as building blocks to model more complex database operations. Vanzyl et al. [12] have investigated the performance of an open source object persistence tool. They have compared the performance of Hibernate, the representative of object-relational mapping tool and db4o, the representative of object-oriented database system. The OO7 benchmark has been used to compare the speed of execution of operations for Hibernate and db4o. Danturthi [13] has presented a comparative analysis of two different database system SQL Server and db4o for the development of web application. UML diagrams have been used for the modeling purpose.

3. PROBLEMS WITH EXISTING DATABASE

The Relation Database Systems (RDBMS) are used in almost every area where data storage is needed. Varieties of database management systems are available in the market. Relational database management system is based on solid mathematical foundation and very powerful to handle a lot of data which helps them to appear everywhere where software systems. However, RDBMS seems extremely inefficient in some new application areas involving processing of large amount of complex data such as image collections, video stream collections etc. These new applications areas require the database system to handle features of complex data types, abstract data structures, data encapsulation, novel methods for indexing and querying. The main weaknesses of RDBMS are listed below:

(A) **Representation of Real World:** The real world contains objects or entities but they are kept in relations like tuples and keys. In relational database if the data is available in two different tables and we want some information from both of them then it requires a join operation.

(B) **Homogeneous data:** The relational data model has limited number of data types which are not sufficient to represent real world objects. The item in the tuple must be an atomic data type because it assumes both horizontal and vertical homogeneity.

(C) **Lack of New Data Type:** New data type can not be added to the system after its development. The DBMS must support new data types such as video, audio, images, biological sequences, etc. These data types have type-specific operations which should be encapsulated with the type.

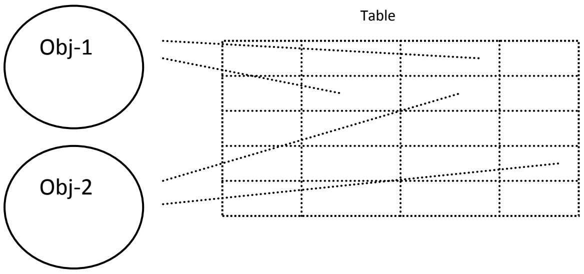

Fig 1: Representing Objects into Table in RDBMS

(D) **Semantic overloading:** The relational data model is not enough to represent data and data relationships. During the normalization process the relation is decomposed into several relations. The decomposition causes slow and complex query processing because the Projection, Selection and Join operations have to be used frequently for the reconstruction the objects.

(E) **Recursive queries and Limited Operations:** Specification and implementation of recursive queries is a difficult task. Computational incompleteness of the SQL is responsible for this problem. Moreover the relational model has a fixed set of SQL operations and new operations are not allowed in the system.

(F) **Impedance mismatch:** The relational model uses mixed different programming paradigms, in which data with different types and locations can be handled at the same time. More importantly, the complex data structures created by application are mismatching with the data type in the database systems.

4. NEED FOR OODBMS

The relational database management is suffering from certain problems caused due to complex and high volume of data. The increased emphasis on process integration has motivated us to use object oriented databases system. Areas of Computer Integrated Manufacturing (CIM), Computer Aided Design (CAD), Computer Aided Manufacturing (CAM), Computer Aided Software Engineering (CASE) and advanced office automation system use object-oriented database system to handle complex graphical data and hypermedia data. Suppose a company has developed several software agents and want to reuse them in different ways. Now, it can be done in two ways: first by reusing the source code and recompile the agent every time and second to store the agents in an object oriented database along with its behaviors. The second approach i.e. using an object oriented database system complete agents with attributes and behavior could be stored and reused. The object-oriented databases can also be used in case of very complex data. Two major aspects due to which object-oriented database system is highly required are as following:

- Handling Complex Data
- High Performance

5. COMPARISION BETWEEN RDBMS AND OODBMS

Main goal of object-oriented database system is to provide encapsulation and class independence features for the objects. The classes can be reorganized without affecting its usage in any application. Main objective relational database system is to reorganize the data physically without affecting its usage. That means it ensures the feature of data independence from application programs.

5.1 Representations used in Data Models

Table-1 Representations used in OODBMS and RDBMS

OODBMS	RDBMS
Object	Entity
Class	Types of entity

Hierarchy of Class	Database Schema
Instance of Class	Tuple
Attribute	Attribute
Relation	Relation
Object ID	Primary Key
Encapsulation	NA
Inheritance	NA

5.2 Conceptual Model

OODBMS uses consistent conceptual model where the classes of objects represents the concept of application. They use a consistent model for the analysis, designing, programming and accessing the database.

On other hand, the RDBMS uses different conceptual model for analysis, designing and programming.

5.3 Storage

OODBMS can store both data and methods. The storage is n the form of active objects which can execute their methods. It also fulfills the encapsulation property where data the stored data can be used through the methods of their classes.

RDBMS can store only data. The stored data is passive in nature as the operations are brought into use when the data is used. Data partitioning is possible depending on user requirements and applications.

5.4 Usage of Data

The data structure used for OODBMS may be complex, which involves different types of data types for sound, image, video etc. To increase the performance of system the concept of chained data is used.

The RDBMS stores data using simple data structures as attribute, tuples and relation. Separate tables are used to store different data and join operators are used to fetch data from to combine the data.

5.5 Redundancy

In OODBMS, The data and methods used are non-redundant which is achieved by using encapsulation and inheritance. In RDBMS, Data non-redundancy is achieved by data normalization which eliminates or reducing data redundancy and the concept s used in the stage of designing the database.

5.6 Independence

Independence of classes is also supported as the reorganization of classes is possible without affecting the mode of their usage. It fulfills the feature of data independence, i.e. the data can be reorganized and modified without affecting next higher level of design.

6. CALCULATION OF RESPONSE TIME

Metrics chosen for the comparative estimation are analysis and calculation of Response Time for various tasks. Firstly, the general analysis based comparison is made between relational database management system and object-oriented database system. In second step response time is calculated for the comparative analysis.

To test the performance a simple class is developed with the name of User having fields UID, Name, Address and MobNumber. The field UID is the primary key and different objects of the User class must have different IDs. 40,000 different customer objects were first inserted into the database, and then the objects were queried back by their IDs.

On other hand, a table is created in Mcrosoft SQL server with five columns having same names as the fields of User class. Then a batch query is written to insert the 40,000 tuples in the Customer table. An equivalent program is used to query the student information back and to pack the information to Student objects.

6.1 Response Time in Writing Objects to Database

Figure 4.10 represents the comparison of response time in writing the objects using DB4O and SQL Server 2008. Performance of SQL Server is better when no of objects are low but for increased situation object-oriented DBMS (DB4O) performs well.

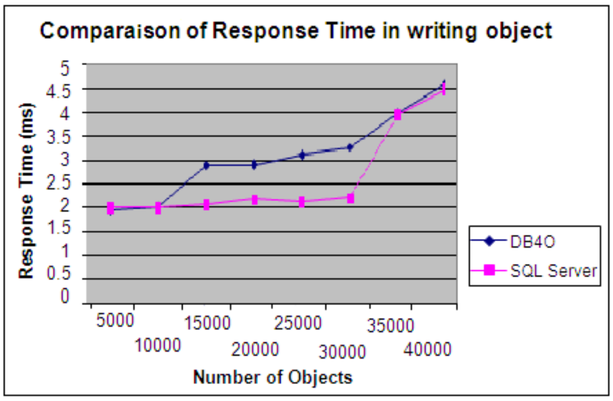

Fig 2: Comparison of Response Time in Writing Objects

6.2 Response Time in Updating Objects to Database

Response time in updating the objects is compared in Figure 4.11 for DB4O and SQL Server 2008. Performance of object-oriented DBMS (DB4O) is better in almost every situation.

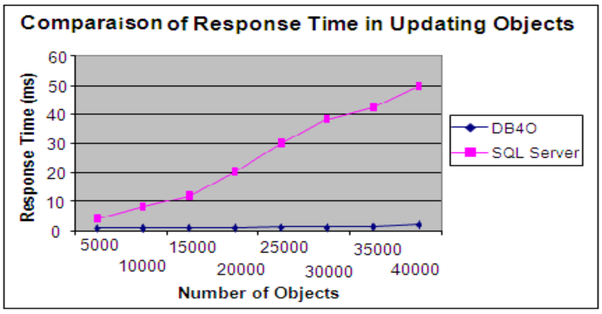

Fig 3: Comparison of Response Time in Updating Objects

6.3 Response Time in Retrieving Objects

Figure 4.12 shows the comparison of response time in retrieving attribute value for objects using DB4O and SQL Server 2008.

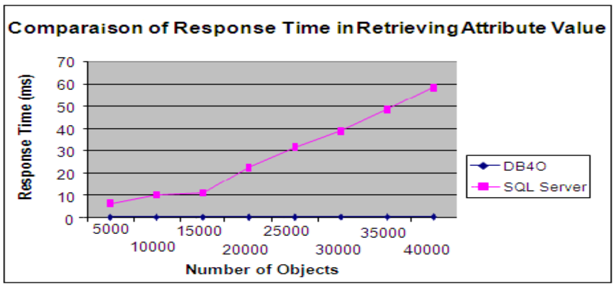

Fig 4: Comparison of Response Time in Retrieving Attribute Value

The result shows that object-oriented DBMS (DB4O) performs well.

7. CONCLUSION AND FUTURE WORK

The framework designed for the performance estimation of object-oriented database system is very effective. The proposed framework focuses on performance analysis of DB4O, comparative computation of DB4O and SQL Server-2008 and UML based performance model. From the work done in first sub section, it is concluded that OO7 benchmark is better as compare to hyper model and OO1. Hardware and software requirements are defined for the performance analysis of DB4O and creation time, insertion time, deletion time and query time is computed.

The comparative analysis part concludes that SQL Server 2008 has shown better response time in writing, updating and retrieving the objects when the number of objects and complexities are low but when the number of objects and complexity increases then DB4O has shown better performance than SQL Server 2008. The third section of present work concludes that the proposed steps for the generation of performance model are very efficient. It involves the transformation of class diagram into equivalent annotated class diagrams, which is used for designing of execution graph and then queuing network based performance model is designed which is very effective. Further, to calculate the quantitative values related to performance, QSolver/1 can be used. Value of workload intensities and other inputs are given to a performance tool and their response times can be calculated.

8. ACKNOWLEDGEMENTS

The authors are thankful to Prof. R.C. Sobti, Vice-Chancellor, B.B. Ambedkar University (A Central University) Lucknow for providing the excellent facility in the computing lab of B. B. Ambedkar University, Lucknow, India. I am also thankful to Mr. Martin Gogolla and Mr. Mark Richters for his support to understand the concepts used in the design and implementation of our project.

9. REFERENCES

[1] H. Boral and D. J. DeWitt, “A methodology for database system performance evaluation,” ACM SIGMOD Record, Vol. 14, No. 2, June 1984.

[2] J. Duhl and C. Damon, “A performance comparison of object and relational databases using the sun benchmark,” Proc. of the Conference on Object-Oriented Systems, Languagesand Applications(OOPSLA), San Diego, California, USA, September 1988 pp. 153-163.

[3] R. G. G. Cattell, “Object-Oriented DBMS Performance Measurements,” Proc. of an International Workshop Advanced Programming Environments, Trondheim, Norway, September 1988, pp. 364-367.

[4] Zdonik, S. B., and Maier, D., 1990. Fundamentals of Object-Oriented Databases, Morgan Kaufman.

[5] J. D. Ullman, “A comparison between deductive and object-oriented database systems,” In [DKM91], December 1991, pp. 263-277.

[6] Ishikawa, H., 1993. Object-Oriented Database System, Springer-Verlag.

[7] A. B. Chaudhri, J. A. McCann and P. Osmon, “A Performance Study of Object Database Management Systems, Theory and Practice of Object Systems,” Vol. 5, No. 4, 1999, pp. 263-279.

[8] N. Gorla, “An Object-Oriented Database Design for Improved Performance,” Journal of Data & Knowledge Engineering, Vol. 37, 2001, pp. 117-138.

[9] Smith, C.U., and Williams, L.G., 2002. Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software, Addison-Wesley.

[10] S. Bagui, “Achievements and Weaknesses of Object-Oriented Databases,” ETH Zurich, Chair of Software Engineering, Journal of Object Technology, Vol. 2, No. 4, July-August 2003.

[11] S. Yin and I. Ray, “Relational database operations modeling with UML,” Proc. of the IEEE 19th International Conference on Advanced Information Networking and Applications, Taipei, Taiwan, March 2005.

[12] P. Vanzyl, G. Derrick, Kourie and A. Boake, “Comparing the Performance of Object Databases and ORM tools,” Proc. of South African Institute for Computer Scientists and Information Technologists, 2006, pp. 1-11.

[13] S. Danturthi, “Comparative Study of Web Application Development with SQL Server and Db4o,” Master's thesis in Computer Science, Mälardalen University, Västerås, Sweden, 2011.