International Journal of Technology Research and Management
ISSN (Online): 2348-9006
Vol 5 Issue 8 August 2018

Impact of Object Oriented Concepts on Database Management System

Sheetal Goutam
Assistant Professor, Pioneer Institute of Professional Studies, Indore
sheetal.goutam@pioneerinstitute.net

Abstract: *There has been emergence of Object-oriented databases as a better alternative to the traditional relational databases. In Object Oriented based Databases capabilities of Object based paradigm for Programming and databases are combined to remove the limitations of Relational databases and on the demand of some advanced applications. Even though they are yet to gain a strong footing commercially, they have laid down a very firm foundation for a database management system that will efficiently cater to the needs of fully automated object oriented database management systems in the future. The paper also compares empirically the performance of OODBMS and RDBMS.DB4O and SQL server have been adopted as the candidate databases for the experiment purpose.*

Keywords: *DBMS, OODBMS, RDBMS, Classes, Objects.*

1. INTRODUCTION

History of data processing goes through many different changes with different technologies along with the time. In decade there is huge increase in the volume of data that need to be processed due to which sometimes old technology do not work and need to come with new technology to process the data. History of database technology has used Unit Records & Punch Card, Punch Card Proliferation, Paper Data Reels, & Data Drums, File Systems, Database Systems, NoSQL and NewSQL databases. From last five decades, the mostly used technology is database management systems.

The evolution of databases can be traced back to 1960's, where disks and drums acted as a means of information storage. After a decade it was felt that data should be made independent of the logic of the application program. The first ever databases were navigational in nature, where record pointers were used to access data by concerned applications. The record pointer was made to point to the record being accessed. This was the predecessor of the IBMs hierarchical model (IMS System) and network model. Following the aforementioned models, came the relational model which emphasized more on the storage content rather than on the links to retrieve the data and till date it is the most widely used type of database. Database Management Systems which is the collection of software or programs to maintain the data records. Initially, two models are proposed are hierarchical and network models, but these models don't get much popularity due to their complex nature. Then a researcher E.F. Codd comes up with a new data model known as relational model in which data items are stored in a table. Many DBMS's are developed on the basis of this model. This is the most popular model till now because it has conceptually foundation from relational mathematics. With the rapidly progressing industrial sector and the growing population the need to store a large variety of data surfaced. Soon it was found that the Relational models were somewhat limiting as they had a very rigid structure which was to a great extent different from the real world scenario, also it provided no support for new types of data such as graphics, XML, 2D and 3D spatial data. In today's world, most of these client server applications use a Relational Database Management System (RDBMS) as their data store and an object oriented programming language as development platform. As a result main programming construct is considered as 'objects' whereas database management system supports storage of data in form of relations and tuples. This brings in the inconsistency between storage model and programming model. i.e objects must be mapped to tuples while performing storage or retrieval of data by the application. This mapping induces indispensable performance penalty.

OODBMS is the remedy to this performance penalty. With the object model now in the picture, the developers aim to reduce the overhead of translating information representation in the database to an application specific depiction. Contrary to the traditional databases (using relational models); an object model facilitates data persistence and storage by storing objects in the databases. The relationships among various objects are ingrained in the structure of the objects. This is mainly applicable for complex data structures such as 2D and 3D graphics which must otherwise be flattened before they are eligible for storage in the relational databases. An OODBMS combines object-oriented programming paradigm with traditional database management principles. That is, it has to inevitably support abstraction, encapsulation, polymorphism and inheritance and at the same time it should provide means for database management concepts such as the ACID properties (Atomicity, Consistency, Isolation and Durability) which lead to system integrity, support for a query language and secondary storage management systems which in turn allow for managing very large amounts of data. Therefore, Classes, Objects, support for class hierarchy, complex objects, persistent object storage, overloading , overriding, secondary storage management, concurrency , recovery are some of the key concepts among others which are to be enforced by a database management system to be called as an OODBMS. This paper discusses many aspects of Object Oriented Database Management System. After introducing the objective of the paper, manuscript talks about the way evolution has taken place in the area of database management system over the period of time.

2. NEED OF OBJECT ORIENTED DATABASE

There are many reasons for need of OODBMS:

- Process Integration: With the automation of many plants, there arises a need for process integration, which can be best handled by OODBMS.
- Complex Structures: Complex object structures like that of CAD/CASE applications which need to store digital circuits in a way that the information remains intact along with its respective operations, we require OODBMS. In no way can RDBMS serve this need.
- Storage: Some applications demand the need for storing classes as well as objects along with associations and methods. OODBMS is the most apt solution for this problem as it inherently stores the data in the form of objects that are nothing but attributes + methods.

Moreover there are a few more reasons of preferring OODBMS over RDBMS

There are three reasons for need of OODBMS:

A. Limitation of RDBMS

B. Need for Advanced Applications

C. Popularity of Object Oriented Paradigm

A. Limitation of RDBMS

These limitations are in relational model. Due to this these limitations are reflected to all RDBMS .

These limitations are:

1. Poor representation of real world entities: The Relational model cannot represent real world in proper way because it has only one semantic that is table which can represent the real world entity in proper way.

2. Normalization is necessary, but sometimes not useful: Normalization in RDBMS to maintain the consistency of the database, but some broken relations is not related to real world.

3. Overloading of semantic structure: Relational Data Model has only one semantic structure for representing data and relationship that is table. Due to this, sometimes it becomes very difficult to find out that which is going to model data or relationship?

4. Poor support for integrity and enterprise constraints: Constraints are very much needed for your database have to be desired data. RDM supports only limited number of constraints. The enterprise constraints are those which are defined by industry standards.

5. Homogeneous data structure: RDM requires homogeneous data structures like:

- RDM assumes both horizontal and vertical homogeneity.
- Relational mathematics algebra has only fixed number of operations due to which Relational Model operations cannot be extended.

6. Tables can store only atomic/single value: No doubt, this is property of RDM. But sometimes in many situations this property becomes its limitation.

7. Normalization is strongly recommended: Most of the situations, you have must normalize the relation make the data consistency inside your database.

8. Difficulty in handling recursive queries: There is very poor support to handle recursive queries in RDBMS.

For this you must know:

- Depth of recursive query must be known.
- You can use the transitive closure operations to

handle recursive queries in RDBMS.

9. Impedance mismatch: SQL Data Manipulation Language (DML) is lack computational completeness . To overcome this situation, you must embed the SQL with any high programming language like C++, Java, and C #. Due to there will be impedance mismatch between two language SQL and higher programming language.

10. Poor support for long duration transactions: In RDBMS, generally transactions are short lived and concurrency control techniques or mechanisms are not good for .long duration transactions

B. Need for Advanced Applications

a) Computer Aided Design (CAD):

- In these types of applications, relevant data about buildings, airplanes and integrated circuit chips is stored and managed. In this type of applications, database design may be very large.
- Design in these types of applications is not static. This design is evolves through the times. Updates need to be propagated.
- These applications require version control and configuration management.
- These applications require complex objects for their development. For example, a car's component may be related to other components.
- Need long duration transactions because sometimes updates are for reaching.
- Support for cooperative engineering because most of the times many people work on same design.

b) Computer Aided manufacturing (CAM):

- These application data is very much similar to CAD, but needs discrete production.
- These applications must respond to real time events.
- Generally algorithms and custom rules are used to respond to a particular situation.

c) Computer Aided Software Engineering (CASE):

- These applications manage data about the phases of software development life cycle.
- Design may be extremely large.
- Involves cooperative work.
- Need to maintain dependencies among components.
- Versioning and configuration management.

d) Network Management Systems:

- Coordinates communication services across the network.
- These systems are used for such tasks as network path management, problem management and
- network planning.

e) Other Applications: The Object Oriented Database also used in Office Information Systems, Multimedia systems, Digital Publishing and Geographic information Systems.

C. Characteristics of OODBMS

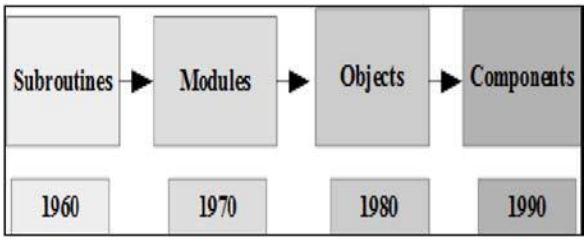
Integrated development system: As mentioned earlier it successfully manages to overcome the problem of "impedance mismatch" hence providing a single development environment in which the data storage model is maps with and equivalent to the programming model.

- Abstraction: It is process of finding important aspects of an entity and ignoring unimportant aspects such as implementation details. The properties comprise two things state and behavior .A state is models through the attributes of object and behavior is models through operations executed on data by object.
- Encapsulation: OODBMS wraps up the object attributes along with the relevant methods in a single unit called the class.
- Object: An object is something uniquely identifiable, models a real world entity and has got state and behavior. The only big difference between entity and object is that entity has only state has no behavior, but object contains both state and behavior. Example: Student.
- Extendable data types: Besides the commonly used data types like text, number, memo etc., OODBMS provides a way of readily handling complex data types like graphics, videos, BLOBs etc with utmost ease.
- Methods and Messages: These implement the behavior of an object and involve encapsulation.
- Message: It is a request or call to an object to execute the method that is defined by message.
- Inheritance: A very useful feature provided by OODBMS for instance; if there are two objects Employee and Manager in RDBMS what one can do is enter the records of both manager and an employee along with their designation to differentiate between the two. In real world even the manager is an employee, but RDBMS stores the data that does not map the real world scenario. On the other hand with OODBMS we can create two classes Employee and Manager and let the Manager derive its primary functionalities from the Employee class through the feature of inheritance.
- Polymorphism: It means "many forms". It is dynamic feature which executes at run time of program. It involves the concept of overriding and overloading.
- Persistence: The object persistence refers to the

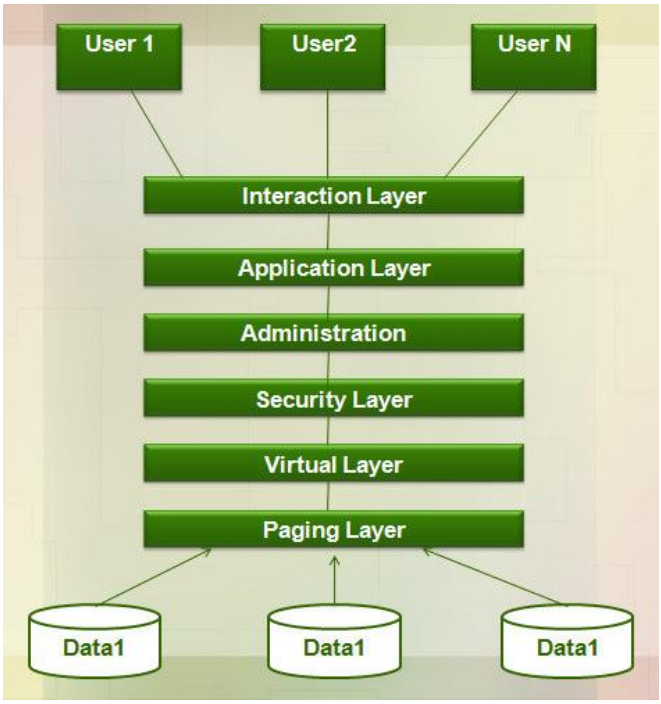
capability of objects to preserve their state across different invocations. If this feature is not present the scope of an object's state lasts until the time application goes out of RAM. If one desires the objects' state to be preserved, persistence is the way.

- Relationships: It is basically an association between two things. These are represented through reference attributes, typically implemented through OID's. Types of binary relationships are:
  - One to One relationship
  - One to Many relationship
  - Many to One relationship
  - Many to Many relationship

The different programming paradigm during the different decades:



3. ARCHITECHTURE OF OODBMS



The architecture of OODBMS is divided into six layers , each with a functionality of its own figure

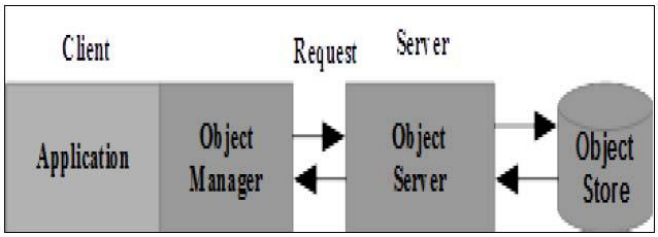
Interaction layer is the top-most layer in the OODBMS architecture; it enables the users to interact with the database. Then follows the Application layer which acts as the interface between the user and the data store. The Administration layer which is the third layer in the architecture controls the flow of data as well as provides the permission to access certain data. The following layer i.e. the Security layer, is responsible to provide the full security to data and also provide to the security of application used to manage the data. The Virtual layer which is the next layer helps in managing large data. The Paging layer is the last layer of the architecture, is responsible for dividing the data into pages so that they can be managed easily.

The basic theme of OODBMS is to add persistence to OOPL as they provide object orientation. The major difference is that here database needs to store data as well as methods

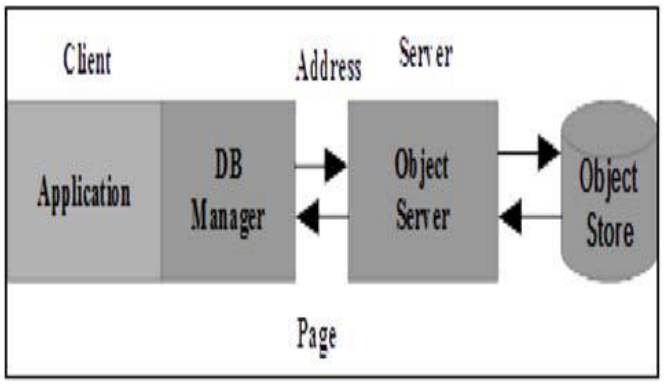
Client/Server Architecture: There are three basic client/server architecture approaches:

a) Object Server
b) Page Server
c) Database Server

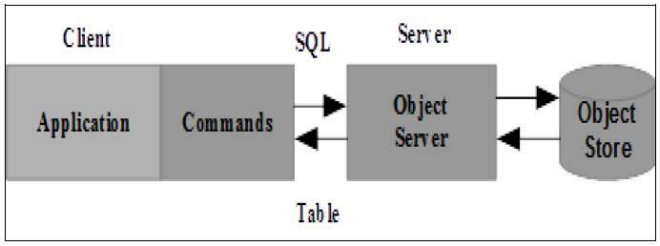
a) Object Server: There is a distributed processing environment between the two parts client and server. Typically, server is responsible for other OODBMs functions. Transaction control management and interface to programming language is handled by Client.



b) Page Server: In this client-server model, client is responsible for database processing most of the times. Secondary storage management and response to requests is handled by server. A page can contain many complex objects or normal objects.

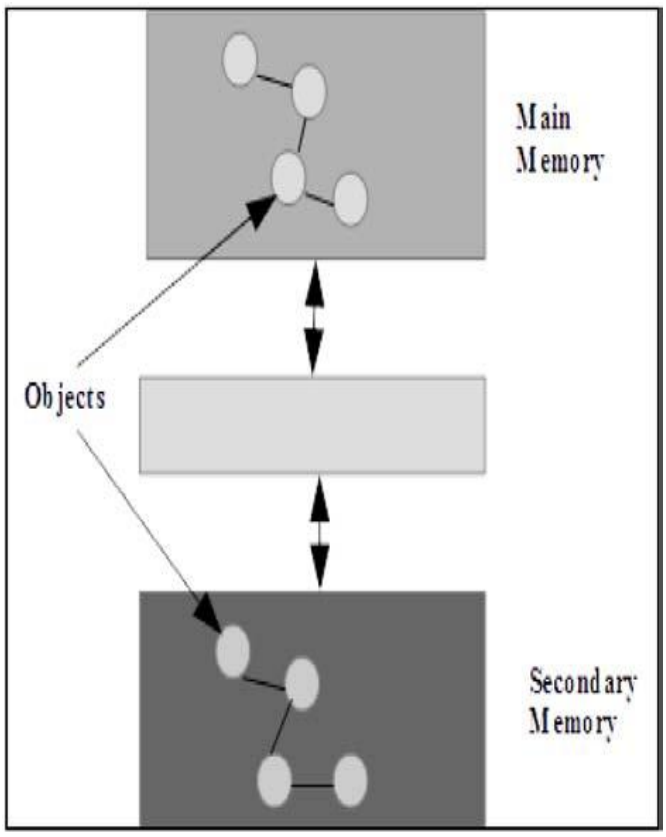


c) Database Server: In this approach, Client simply passes the request to the server, receives results and passes them to application. Most of database processing done at server .This approach is used mainly by RDBMS's.



4. OODBMS PERSPECTIVES

Memory Management in OODBMS: There are different memory management techniques for different OODB systems. For Extended Relational Model database Systems and Object/Relational (by definition) Database Systems (ORDBMS) Two Level Storage Model is used and for pure OODBMS is Single Level Storage Model is used.



5. EXPERIMENTS AND RESULTS

For comparing both the databases empirically, instances of SQL server as RDBMS and DB4O as OODBMS were used. Here, for the sake of completion we present the procedure for installing the instance of DB4O.First step is to install a plug-in called DB4O. The DB4o distribution includes Object Manager Enterprise (OME) which allows you to browse your DB4O database in a graphical interface [10]. With OME you can check which objects were stored and examine the database structure. Using the OME ad-hoc query builder, you can create queries to validate business logic in your application or handle simple reporting requirements.

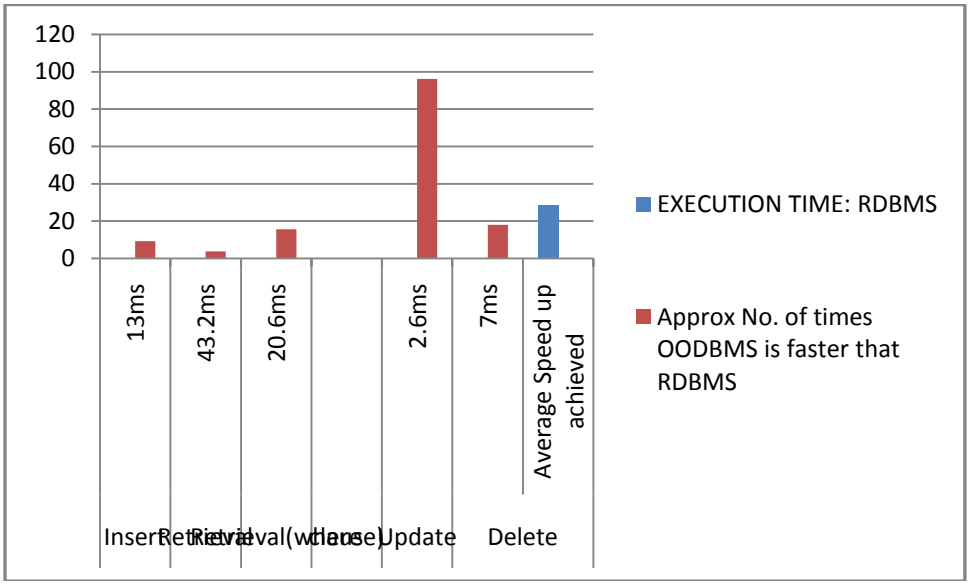
The step that follows next is adding the said plug-in to Eclipse IDE and the following are the steps to do so:

- create a folder named "lib" under your project directory, if it doesn't exist yet.
- copy db4o-*.jar to this folder.
- Right-click on your project in the Package Explorer and choose "refresh".

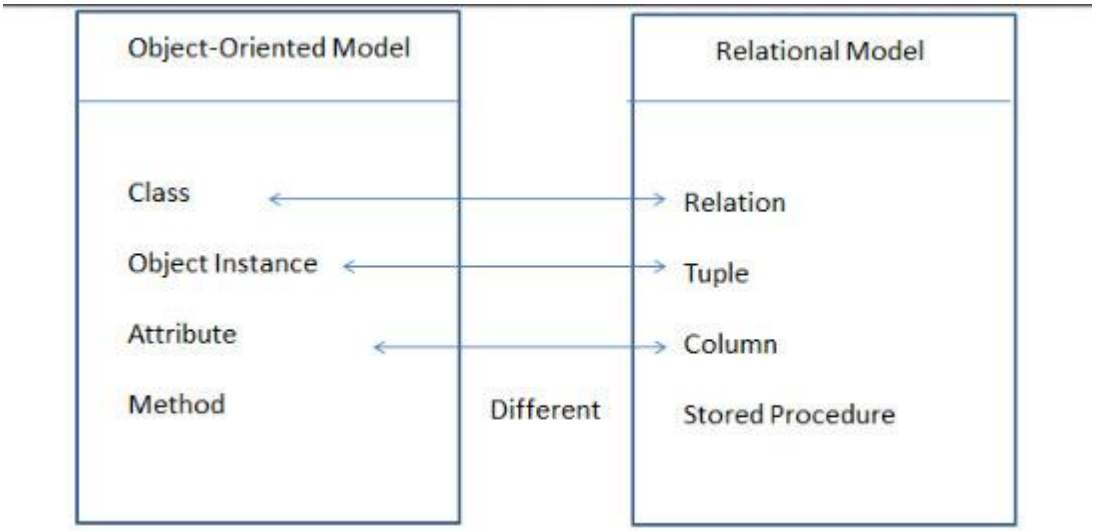
- Right-click on your project in the Package Explorer again and choose "properties".
- select "Java Build Path" in the tree view on the left.
- select the "Libraries" tab page.
- click "Add Jar".
- the "lib" folder should appear below your project.
- choose db4o-*.jar in this folder.

Insert, simple select , select with where clause, update and delete were fired on both the databases.

QUERY	EXECUTION TIME: OODBMS	EXECUTION TIME: RDBMS	Approx No. of times OODBMS is faster that RDBMS
Insert	13ms	120ms	9.23
Retrieval	43.2ms	165ms	3.81
Retrieval(where clause)	20.6ms	321ms	15.58
Update	2.6ms	250ms	96.15
Delete	7ms	126ms	18
Average Speed up achieved			28.55



Graphical Representation of the Execution time taken by various queries on both the databases



Points on which RDBMS and OODBMS Differ

6. CONCLUSION

As No doubt, relational databases are very popular and there are found everywhere. The object oriented database comes into action in mid-1985's to remove the limitations and to support some advanced database applications like CAD, CASE etc. Another point that provides the momentum to develop object based database is popularity of object based programming. But through this study we have found that, Object-oriented database surpasses traditional DBMS in many aspects yet it still lacks popularity in the commercial sector because of certain issues that is faced by OODBMS like it lacks maturity. There is a need for better tools to implement it, and also requires the establishment of certain standards. OODBMS's removes the limitations of RDBMS's, also provide support for advanced database application with some additional features. But due to the lack of standards, they do not get much popularity in the industry. Then after some time, some limitations are found in object oriented database management systems Currently OODBMS and RDBMS are working supplementary to each other leading to a new concept called the ORDBMS. But in the near future OODBMS will become the new standard for the database.

REFERENCES

[1] Object Oriented Database Systems: Approaches and Architectures by C.S.R. Prabhu.
[2] Database System: A Practical Approach to Design, Implementation and Management by T. Connoly and C. Begg.
[3] Saxena, Vipin, and Ajay Pratap. "A Framework for Performance Estimation of Object-Oriented Databases".
[4] Gorla, Narasimhaiah. "An object-oriented database design for improved performance". Data & Knowledge Engineering
[5] http://marketplace.eclipse.org/
[6] http://www.db4o.com
[7] Atkinson M., Bancilhon F. ,Dewitt D. ,Dittrich K., Maier D. and Zdonik S. "The Object Oriented Database Manifesto" December 1989
[8] Stefik, M. and Bobrow, D.G, "Object-oriented programming: Themes and variations", The AI Mag., Jan 1986.