A Comparison of Open Source Object Oriented Database Products

Peter Khayundi[1] Jim Chadwick

Department of Computer Science, University of Fort Hare

pkhayundi@ufh.ac.za jchadwick@ufh.ac.za

[1] All correspondence should be addressed to this author – TEL/FAX 0406022464

Programme Topic: Network Services, Electronic Commerce

Abstract – Object oriented databases have been gaining popularity over the years. The reliance on relational database management systems for data needs has come under threat from object oriented databases. The ease of use and the advantages that are offered by object oriented databases over relational databases has made them a popular choice amongst database administrators. Though object oriented databases may not, in the long term, replace the more traditional relational databases, it is hoped that improvements in operations and workings of object databases will improve their reliability and ability to handle everyday data needs. This study compares some open source object oriented database products available. Various standard operations are carried out on each product and the time taken to complete each operation is calculated. The results would be of value to practitioners of electronic commerce who are considering the use of object oriented databases.

Keywords – Object oriented databases, open source.

I. INTRODUCTION

Object database technology has been preferred by database developers because it is an ideal match for object oriented environments like Java and .NET [1]. Previously, developers had to rely on using relational databases and object oriented languages. This led to the "object-relational (OR) mismatch", with relational databases being incompatible with object oriented programming languages. The emergence of object oriented databases has solved the object-relational mismatch by allowing developers to create databases using object oriented programming languages such as Java and .NET.

The introduction of object databases was seen to provide a facility to support richer data types, a feature which was not provided for in relational databases. Relational databases are still superior when it comes to performing queries on data using SQL statements, but object databases also use a query language, object query language (OQL), which is similar to SQL [4].

Object databases can be represented as either a pure object oriented database management system, or as a hybrid, or post-relational, database management system. The first type is used in many mission-critical applications. Some examples of companies using them include Adidas AG, the Federal Aviation Authority (FAA), and Sales Media, Inc [3].

This paper looks at various object oriented database products available. The products are open source databases, namely db4o [1], Neodatis [7], and Perst [5].

The objective of the study is to compare the performance of these products in carrying out the standard operation of creating database objects and calculating the time taken to complete this operation. The other standard operations of searching, updating, and deleting database objects will be investigated further in future work.

II. MATERIALS AND METHODS

A. *DATABASES*

In order to evaluate the databases for this study, some criteria were applied in selecting them. They were open source databases written in Java, had good documentation that accompanied them as well as sample code, and support in the form of forums on their websites. The products selected were:

i) *Db4o*, an open source object oriented database by db4objects, Inc. The product website offers support in the form of forums, downloads, and subscription to a newsletter.

ii) *Neodatis*, an open source object oriented database developed by Olivier Smadja, Cristi Ursachi, and Marcelo Mayworm. This database is easy to use and can be used as an embedded database or in client/server mode. It also is supported by a forum.

iii) *Perst*, an open source object oriented database by McObject LLC. This database offers a very small data footprint, consisting of five thousand lines of code, making it compact and fast [6]. Support is also provided, with a section on the website offering links to forums, contact support, downloads, and frequently asked questions (FAQs) [5].

B. *Experiments*

The aim of the experiments was to record the amount of time it took to create a certain number of objects in each database. Further experiments will be conducted to record the times taken for updating and deleting objects in the databases, as well as performing queries on data.

The times were recorded by collecting timing data within the database code. The process was repeated a number of times in order to collect the time taken to create different numbers of objects, ranging from 5 to 50. This particular range of objects was chosen because it was able to show how each database was able to handle the creation of this range of objects. Further experiments will be conducted with larger ranges of objects so as to record the behavior of each database.

III. Results

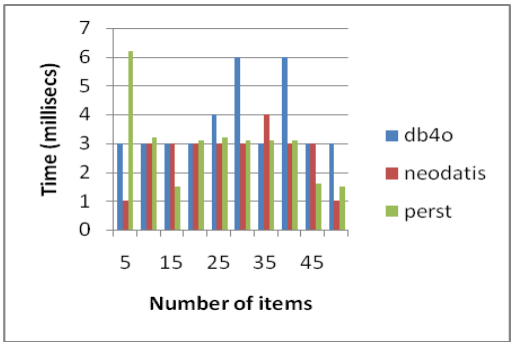

Fig. 1. Timing data for the creation of database objects using Db4o, Neodatis, and Perst. The objects range in size from 5 to 50 and the times are calculated in milliseconds.

The graph in Fig. 1 was generated from data collected when objects of various sizes were created in the databases under experiment. These objects were created and inserted into the various databases, with the times taken being recorded in milliseconds. The times were then tabulated and a graph was created.

When objects were first created, the time taken in Perst was very high compared to the other databases. But this time became less as more objects were created. Db4o recorded higher times as objects were added, with the highest times being seen when adding between 25 and 45 objects. With Neodatis, the times varied from the other two databases.

According to the graph, it took longer to create objects in Db4o than in Neodatis or Perst. Therefore, it can be concluded that Perst was the fastest database when it came to creation of objects, followed by Neodatis, with Db4o coming a distant third.

IV. Conclusion

The methods that were used to evaluate the databases for timing purposes proved to be effective. Further experiments will be carried out on the databases to evaluate their response time in relation to updating, performing queries and deletion of data. The data collected will be used to determine which database is better suited for a particular operation.

V. References

[1] Db4o website: http://www.db4o.com

[2] Db4o product and company information. Available: http://www.db4o.com/about/company/backgrounder/db4objects%20Business%20Backgrounder%20November%202005.pdf

[3] Harrington J.L. 2000. *Object-Oriented Database Design Clearly Explained*. London: Academic Press

[4] Jackson M. "Thirty years (and more) of databases", *Information and Software Technology Volume 41*, Issue 14, 5 November 1999, Pages 969-978. Available: http://www.sciencedirect.com/science/article/B6V0B-3Y0J9GV-8/1/8b642768cb80c336248e9b606108d83c

[5] Perst website: http://www.mcobject.com/perst/

[6] Perst database introduction: http://www.mcobject.com/perst/perstintro.shtml

[7] Neodatis website: http://www.neodatis.org/

Peter Khayundi was born in Nairobi, Kenya on the 13th July 1981. He completed his undergraduate degree in 2005 and his Honours in 2006. He is currently pursuing his M.Sc. in Computer Science at the University of Fort Hare.