Experiencing with 007 Benchmark for Concurrency Control Technique Performance Evaluations

Woochun Jun and Le Gruenwald
Dept. of Computer Science
Univ. of Oklahoma
Norman, OK 73019

Abstract

The 007 benchmark provides a comprehensive test of OODB (Object-Oriented Data Base) performance. In this paper, we discuss the problems and the possible suggestions for the 007 Benchmark, based on our experience, in terms of performance evaluation of OODB concurrency control techniques. Our discussions are focused on three access types of OODB concurrency control techniques: conflict among methods, class hierarchy locking and nested method invocations.

1. Introduction

Performance is a major concern for the developers of OODB systems. But, in general, measuring OODB performance is difficult since each application may need different requirements. The typical way of performance test is to select a benchmark and run it on several systems so that performance can be compared. There have been a number of benchmarks in OODB environments ([Catt,1992],[Ande,1992],[Berr,1992]). The 007 benchmark ([Care,1993],[Care,1994-1],[Care,1994-2]) provides a comprehensive test of OODB performance. Especially, it provides wide range of pointer traversal, a rich set of updates and queries.

We choose the 007 benchmark in order to evaluate our new locking_based concurrency control technique for OODB. For performance comparison, we also select two existing locking_based concurrency control techniques, called Orion [Garz,1988] and Malta's ([Malt,1991],[Malt,1993]), respectively. In general, locking_based concurrency control schemes for OODBs have three access types: conflict among methods, class hierarchy locking and nested method invocations [Jun,1997]. We have been testing all three techniques for each access type.

In the paper, we report the problems of 007 benchmark during OODB concurrency control technique testing and suggest possible solutions or modifications on 007 benchmark. Especially, we discuss problems of the 007 benchmark, in terms of structures and operation (or transaction) types for each access type.

The remainder of the paper is organized as follows. In Section 2, we give the brief description of 007 benchmark. Section 3 describes problems associated with the 007 benchmark for concurrency control technique testing for each access type. Finally, in Section 4, we give the possible suggestions for 007 benchmark.

2. The 007 Benchmark Description

There are three sizes of the 007 benchmark: *small*, *medium* and *large*. We select small size for simplicity in implementation. There are ten classes in the 007 benchmark. Among those ten classes, classes DesignObj and Assembly serve as abstract superclass in which provide class definitions but not instance object. The DesignObj is the root of the class hierarchy and is (direct) superclass of classes AtomicPart, CompositePart, Assembly and Module, respectively. Also, the Assembly class is (direct) superclass of classes ComplexAssembly and BaseAssembly, respectively.

The 007 benchmark consists of two components: the design library and assembly hierarchy. The key component of the design library is a set of composite parts, forming CompositePart class. Each composite part is associated with document object (Document class). Also, each composite part consists of a set of atomic parts, forming AtomicParts class. In small 007, 20 atomic parts form a composite part. The connections between atomic parts are supported by the a Connection object between each pair of atomic parts.

The Assembly Hierarchy provides higher structure to the Design Library. Especially, each assembly is either consisted of composite part (the assembly is called a BaseAssembly class) or it is consisted of other assembly objects (the assemby is called a ComplexAssembly class). There are 7 levels in the assembly hierarchy. The bottom level of the assembly hierarchy consists of base assembly objects. Each base assembly object is associated with composite part object bi-directionally. The higher level consists of complex assemblies. Each complex object is associated with either base assemblies (if the complex object has level two) or other complex object (if the complex object has higher level). Each assembly hierarchy forms a module which is the largest unit. Each module is associated with a Manual object.

3. Problems with 007 Benchmark

3.1. Overview

In OODBs, a database is a collection of classes and instances where classes and instances are called *objects*. Also, an object participates in various forms of hierarchies among objects such as class hierarchy or class composition hierarchy. The class hierarchy is based on *inheritance* property. That is, a subclass inherits definitions defined on its superclasses. Also, there is an is-a relationship between a subclass and its superclasses. Thus, an instance of a subclass is a specialization of its superclasses (and conversely, an instance of a superclass is a generalization of its subclasses) [Garz,1988]. This inheritance relationship between classes forms a class hierarchy. On the other hand, the class composition hierarchy is based on nested structures in OODBs. That is, an object can be composed of complex objects or atomic object. For example, an object *vehicle* can consists of three atomic objects (say, *id*, *color*, and *drivetrain*) and a complex object *manufacturer*.

Due to hierarchies in OODBs, there are three access types: conflict among methods, class hierarchy locking and nested method invocations. The first access type, conflict among methods, is concerned with access to a single object. In general, there are two types of access to an object: *instance access* and *class definition access* [Cart,1990]. An instance access consists of consultations and modifications of attribute values in an instance or a set of instances. A class definition access includes consulting class definition, adding/deleting an attribute or a method, changing the implementation code of a method or changing the inheritance relationship between classes. In the second access type, an access is concerned with class hierarchy. That is, due to inheritance, while a class and its instance are being accessed, the definitions of the class' superclasses should not be modified. Also, due to *is-a* relationship between classes, the search space for a query against a class, say C, may include the instances of all classes in the class hierarchy rooted at C as well as all instances of C. The third access type is concerned with class composition hierarchy. Due to nested structure of an object, it is natural that a method on a class may invoke another method on its subobject (well call nested method invocation).

3.2. Problems with 007 benchmark for concurrency control scheme testing

For the test case of conflict among methods, we need to test three different types of conflicts: conflict among instance accesses, conflict among instance access and class definition access, and conflicts among class definition accesses. The 007 benchmark provides the structural modification transaction types. But, they are concerned with only instance insertion/deletion. Actually, we need two types of transactions: class definition read transaction and class definition write transactions. Also, for Malta's scheme evaluation, each class needs to have many instance access methods, especially instance write methods so that commutativity relationships among methods can be established. Actually, we suffered from lack of instance write methods.

Second, for the test of class hierarchy locking scheme, we need to test two existing class hierarchy locking schemes: *implicit locking* and *explicit locking*. In implicit locking, locking overhead can be reduced for access to classes near root in the class hierarchy. On the other hand. explicit locking incurs less overhead for access to classes near leaf level. Thus, in order to make difference between two techniques, a class hierarchy needs to have higher level of depth. Unfortunately, the 007 benchmark provides only three levels of depth so that we have difficulties to test two techniques.

Finally, for the test of nested method invocations, we need more instance write methods in order to evaluate our technique, as in the test case of conflict among methods. Also, in the 007 benchmark, the fan-out of each nested method (i.e., the number of method invocations to subobjects) is 3. We feel that this is small to utilize the parallelism between parent/children.

4. Discussions

The 007 benchmark provides a comprehensive test of OODB performance. In this paper, we focused on OODB concurrency control technique testing. We discussed the problems associated with the 007 benchmark in terms of concurrency control technique performance evaluation. In order to provide wide range of tests for concurrency control technique evaluation, we feel that the following are necessary: class definition read and class definition write transactions, more instance access (instance read and instance write) methods per class, deeper class hierarchy and bigger fan-out for the nested method.

References

[Ande, 1990]. T. L. Anderson, et al, "The HyperModel Benchmark", Proc. of the 2nd Int. Conf. on Extending Database Technology, Lecture Notes on Computer Science 416, Springer-Verlag, Berlin, pp. 317 - 331.

[Berr,1991]. A. J. Berre and T. L. Anderson, "The HyperModel Benchmark for evaluating object-oriented databases, In Object-Oriented Databases with Applications to CASE, Networks, and VLSI CAD, edited by R. Gupta and E. Horowitz, Englewood Cliffs, Jew Jersey, Prentice-Hall, pp. 75 - 91.

[Care,1993] M. J. Carey, D. J. Dewitt and J. F. Naughton, "The 007 Benchmark", Proc. of the 1993 ACM SIGMOD Conference on Management of Data, Washington D.C., May, 1993, pp. 12 - 21.

[Care, 1994-1] M. J. Carey, D. J. Dewitt, C. Kant and F. Naughton, "A Status Report on the 007 OODBMS Benchmarking Effort", Proc. of OOPSLA, Portland, Oregon, 1994, pp. 414 - 426.

[Care,1994-2] M. J. Carey, D. J. Dewitt and J. F. Naughton, "The 007 Benchmark", Tech. Report, Dept. of Computer Science, Univ. of Wisconsin, 1994.

[Cart,1990] M. Cart and J. Ferrie, "Integrating concurrency control into an object-oriented database system", *2nd Int. Conf. on Extending Data Base Technology*, Venice, Italy, Mar. 1990, pp. 363 - 377.

[Catt,1992] R. Cattell and J. Skeen, "Object operations benchmark", ACM Transactions on Database Systems, Vol. 17, No. 1, Mar. 1992

[Garz,1988]. J. F. Garza and W. Kim, "Transaction Management in an Object-Oriented Database System", *ACM SIGMOD Int. Conf. on Management of Data*, Chicago, Illinois, Jun. 1988, pp. 37 - 45.

[Jun,1997] W. Jun, An Integrated Concurrency Control in Object-Oriented Database Systems, Ph.D. Dissertation, Dept. of Computer Science, Univ. of Oklahoma, 1997.

[Kim,1990] W. Kim, *Introduction to Object-Oriented Databases*, The MIT Press, 1990

[Malt,1991] C. Malta and J. Martinez, "Controlling Concurrent Accesses in an Object-Oriented Environment", *2nd Int. Symp. on Database Systems for Advanced Applications*, Tokyo, Japan, Apr. 1991, pp. 192 - 200

[Malt,1993] Carmelo Malta and Jose Martinez, "Automating Fine Concurrency Control in Object-Oriented Databases", *9th IEEE Conf. on Data Engineering*, Vienna, Austria, Apr. 1993, pp. 253- 260.