

Object Database Benchmarks

Jerôme Darmont

Universite Lumiere Lyon 2, France

INTRODUCTION

The need for performance measurement tools appeared soon after the emergence of the first Object-Oriented Database Management Systems (OODBMSs), and proved important for both designers and users (Atkinson & Maier, 1990). Performance evaluation is useful to designers to determine elements of architecture and more generally to validate or refute hypotheses regarding the actual behavior of an OODBMS. Thus, performance evaluation is an essential component in the development process of well-designed and efficient systems. Users may also employ performance evaluation, either to compare the efficiency of different technologies before selecting an OODBMS or to tune a system.

Performance evaluation by experimentation on a real system is generally referred to as benchmarking. It consists in performing a series of tests on a given OODBMS to estimate its performance in a given setting. Benchmarks are generally used to compare the global performance of OODBMSs, but they can also be exploited to illustrate the advantages of one system or another in a given situation, or to determine an optimal hardware configuration. Typically, a benchmark is constituted of two main elements: a workload model constituted of a database and a set of read and write operations to apply on this database, and a set of performance metrics.

BACKGROUND

Object Database Benchmarks Evolution

In the sphere of relational DBMSs, the Transaction Performance Processing Council (TPC) issues standard benchmarks, verifies their correct application and regularly publishes performance test results. In contrast, there is no standard benchmark for OODBMSs, even if the more popular of them, OO1, HyperModel, and OO7, can be considered as *de facto* standards.

OO1, also referred to as the “Cattell Benchmark” (Cattell, 1991), was developed early in the nineties when there was no appropriate benchmark for engineering ap-

plications such as computer aided design (CAD), computer aided manufacturing (CAM), or software engineering (SE). OO1 is a simple benchmark that is very easy to implement. A major drawback of this tool is that its workload model is too elementary to measure the elaborate traversals that are common in many types of object-oriented applications.

The HyperModel Benchmark (Anderson et al., 1990), also referred to as the Tektronix Benchmark, possesses a richer workload model than OO1. This renders it potentially more effective than OO1 in measuring the performance of engineering databases. However, this added complexity also makes HyperModel harder to implement.

OO7 (Carey, Dewitt & Naughton, 1993) reuses the structures of OO1 and HyperModel to propose a more complete benchmark and to simulate various transactions running on a diversified database. It has also been designed to be more generic than its predecessors and to correct some of their known weaknesses. However, OO7 is even harder to implement than HyperModel.

OO1, HyperModel, and OO7, though aimed at engineering applications, are often viewed as general-purpose benchmarks. However, they feature relatively simple databases and are not well suited for other types of applications such as financial, telecommunication, and multimedia applications (Tiwary, Narasayya & Levy, 1995). Hence, many benchmarks were developed to study particular domains, such as client-server architectures (Schreiber, 1994), object clustering (Bancilhon, Delobel & Kanellakis, 1992; Darmont, Petit & Schneider, 1998; Gerlhof et al., 1996), object-relational systems (Carey, Dewitt & Naughton, 1993; Lee, Kim & Kim 2000), active databases (Zimmermann & Buchmann, 1995), workflow management (Bonner, Shrufi & Rozen, 1995), CAD applications (Kempe et al., 1995), or the study of views in an object-oriented context (Kuno & Rundensteiner, 1995). A fair number of these benchmarks are more or less based on OO1, HyperModel, or OO7.

An alternative to very specific benchmarks resides in generic and tunable benchmarks such as OCB (Darmont & Schneider, 2000). The flexibility and scalability of OCB is achieved through an extensive set of parameters that helps OCB simulate the behavior of the *de facto* standards in object-oriented benchmarking. Furthermore, OCB’s generic model can be implemented within an object-rela-

tional system easily and most of its operations are relevant for such a system. Hence, it can also be applied in an object-relational context with few adaptations.

Finally, OCB has been recently extended to become the Dynamic Object Evaluation Framework (DOEF), which introduces a dynamic component in the workload (He & Darmont, 2003). Changes in access patterns indeed play an important role in determining the efficiency of a system or of key performance optimization techniques such as dynamic clustering, prefetching, and buffering. However, all previous benchmarks produced static access patterns in which objects were always accessed in the same order repeatedly. In contrast, DOEF simulates access pattern changes using configurable styles of change.

Issues and Tradeoffs in Benchmarking

Gray (1993) defines four primary criteria to specify a good benchmark: (1) *relevance*: it must concern aspects of performance that appeal to the largest number of potential users; (2) *portability*: it must be reusable to test the performances of different OODBMSs; (3) *simplicity*: it must be feasible and must not require too many resources; and (4) *scalability*: it must be able to be adapted to small or large computer systems or new architectures. Table 1 summarizes the characteristics of the main existing benchmarks according to Gray’s criteria. It is important to note that these four criteria are in mutual conflict. For instance, the size and complexity of a relevant workload may come in conflict with its feasibility and possibly with portability requirements. Hence, it is necessary to find the right compromise regarding given needs.

The *de facto* standards in OODBMS benchmarking all aim at being generic. However, they all incorporate database schemas that are inspired by structures used in engineering software, which finally tailors them to the study of these particular systems. Adapting these benchmarks to another domain requires some work and a derived benchmark that takes into account specific elements often needs to be designed. Hence, their relevance decreases when they are applied in other domains but engineering. A solution to this problem is to select a generic benchmark that can be tailored to meet specific needs. However, there

is a price to pay. Genericity is achieved with the help of numerous parameters that are not always easy to set up. Thus, the effort in designing a specific benchmark must be compared to the parameterization complexity of a generic benchmark.

There is also another, very different, “qualitative” aspect of benchmarking that has never really been considered in published benchmarks and that is important for a user to consider when selecting an OODBMS. Atkinson et al. (1992), Banerjee and Gardner (1995), and Kempe et al. (1995) all insist on the fact that system functionality is at least as important as raw performances. Hence, criteria concerning these functionalities should be worked out.

Finally, there is an issue that is not a scientific one. Carey, Dewitt and Naughton (1993) and Carey et al. (1994) pointed out serious legal difficulties in their benchmarking effort. Indeed, OODBMS vendors are sometimes reluctant to see benchmark results published. However, designing relevant benchmarks remains an important task and should still be carried out to help researchers, software designers or users evaluate the adequacy of any prototype, system or implementation technique in a particular environment.

CONCLUSION AND FUTURE TRENDS

The development of new object database benchmarks is now scarce, mainly because the first generation of OODBMSs failed to achieve any broad commercial success. This failure is largely due to the never-ending issue of poor performance compared to relational DBMSs, which are well optimized and efficient. However, with the development of object-oriented programming both off-line and online, the need for persistent objects remains. Object-relational systems are now used more and more frequently (to store XML documents, for instance). Thus, the experience that has been accumulated when designing object-oriented database benchmarks could be reused in this context. The challenge for object store designers is now to produce efficient systems, and sound benchmarks could help them achieve this goal.

Table 1. Comparison of existing benchmarks with Gray’s criteria

	Relevance	Portability	Simplicity	Scalability
OO1	--	++	++	-
HyperModel	+	+	-	--
OO7	++	+	--	-
OCB	++	+	-	++

Strong point: + Very strong point: ++ Weak point: - Very weak point: --

2 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/object-database-benchmarks/14575

Related Content

Evolution of Post-Secondary Distance Education

Iwona Miliszewska (2009). *Encyclopedia of Information Science and Technology, Second Edition* (pp. 1471-1476).

www.irma-international.org/chapter/evolution-post-secondary-distance-education/13771

Understanding a Revolutionary and Flawed Grand Experiment in Blockchain: The DAO Attack

Muhammad Izhar Mehar, Charles Louis Shier, Alana Giambattista, Elgar Gong, Gabrielle Fletcher, Ryan Sanayhie, Henry M. Kimand Marek Laskowski (2019). *Journal of Cases on Information Technology* (pp. 19-32).

www.irma-international.org/article/understanding-a-revolutionary-and-flawed-grand-experiment-in-blockchain/216950

Newborn Verification Using Headprint

Shrikant Tiwariand Sanjay Kumar Singh (2012). *Journal of Information Technology Research* (pp. 15-30).

www.irma-international.org/article/newborn-verification-using-headprint/72708

Evaluation of High School Websites Based on Users: A Perspective of Usability and Performance Study

Ana Maria Santos, José Antonio Cordón-Garcíaand Raquel Gómez Díaz (2019). *Journal of Information Technology Research* (pp. 72-90).

www.irma-international.org/article/evaluation-of-high-school-websites-based-on-users/224980

Eliciting the Requirements for Intelligent Systems in Law Enforcement

Jeffrey L. Goldberg, Sanjeev Kapoor, Corey Wineman, William Carneyand Dorothy G. Dologite (2004). *Annals of Cases on Information Technology: Volume 6* (pp. 418-439).

www.irma-international.org/chapter/eliciting-requirements-intelligent-systems-law/44590