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 \textit{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 applications 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 \textit{de facto} standards in object-oriented benchmarking. Furthermore, OCB’s generic model can be implemented within an object-rela-
Related Content

Temporal Analysis of Information Technology Chargeback Systems
[www.igi-global.com/article/temporal-analysis-information-technology-chargeback/51048?camid=4v1a](www.igi-global.com/article/temporal-analysis-information-technology-chargeback/51048?camid=4v1a)

Leveraging Objects for Mission-Critical Applications
[www.igi-global.com/chapter/leveraging-objects-mission-critical-applications/33494?camid=4v1a](www.igi-global.com/chapter/leveraging-objects-mission-critical-applications/33494?camid=4v1a)

‘Stream of Training’ Approach in Project Management Training
[www.igi-global.com/chapter/stream-training-approach-project-management/54513?camid=4v1a](www.igi-global.com/chapter/stream-training-approach-project-management/54513?camid=4v1a)

Using Agent Technology for Company Knowledge Management
[www.igi-global.com/article/using-agent-technology-company-knowledge/1272?camid=4v1a](www.igi-global.com/article/using-agent-technology-company-knowledge/1272?camid=4v1a)