Database Benchmarks

Jérôme Darmont
ERIC, University of Lyon 2, France

INTRODUCTION

Performance measurement tools are very important, both for designers and users of Database Management Systems (DBMSs). *Performance evaluation* is useful to designers to determine elements of architecture, and, more generally, to validate or refute hypotheses regarding the actual behavior of a DBMS. 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 a DBMS, or to tune a system.

Performance evaluation by experimentation on a real system is generally referred to as benchmarking. It consists of performing a series of tests on a given DBMS to estimate its performance in a given setting. Typically, a *benchmark* is constituted of two main elements: a database model (conceptual schema and extension), and a workload model (set of read and write operations) to apply on this database, following a predefined protocol. Most *benchmarks* also include a set of simple or composite performance metrics such as response time, throughput, number of input/output, disk or memory usage, and so forth.

The aim of this article is to present an overview of the major families of state-of-the-art database benchmarks, namely, relational benchmarks, object and object-relational benchmarks, XML benchmarks, and decision-support benchmarks; and to discuss the issues, tradeoffs, and future trends in database benchmarking. We particularly focus on XML and decision-support benchmarks, which are currently the most innovative tools that are developed in this area.

BACKGROUND

Relational Benchmarks

In the world of relational DBMS benchmarking, the *Transaction Processing Performance Council (TPC)* plays a preponderant role. The mission of this non-profit organization is to issue standard benchmarks, to verify their correct application by users, and to regularly publish performance tests results. Its benchmarks all share variants of a classical business database (*customer-order-product-supplier*) and are only parameterized by a scale factor that determines the database size (*e.g.*, from 1 to 100,000 GB).

The *TPC* benchmark for transactional databases, TPC-C (TPC, 2005a), has been in use since 1992. It is specifically dedicated to On-Line Transactional Processing (OLTP) applications, and features a complex database (nine types of tables bearing various structures and sizes), and a workload of diversely complex transactions that are executed concurrently. The metric in TPC-C is throughput, in terms of transactions.

There are currently few credible alternatives to *TPC-C.* Although, we can cite the Open Source Database Benchmark (OSDB), which is the result of a project from the free software community (SourceForge, 2005). OSDB extends and clarifies the specifications of an older benchmark, AS^{3}AP. It is available as free C source code, which helps eliminate any ambiguity relative to the use of natural language in the specifications. However, it is still an ongoing project and the benchmark's documentation is very basic. AS^{3}AP's database is simple: it is composed of four relations whose size may vary from 1 GB to 100 GB. The workload is made of various queries that are executed concurrently. OSDB's metrics are response time and throughput.

Object-Oriented and Object-Relational Benchmarks

There is no standard benchmark for object-oriented DBMSs. However, the most frequently cited and used, OO1 (Cattel, 1991), HyperModel (Anderson, Berre, Mallison, Porter, & Schneider, 1990), and chiefly *OO7* (Carey, DeWitt, & Naughton, 1993), are *de facto* standards. These benchmarks mainly focus on engineering applications (*e.g.*, computer-aided design, software engineering). They range from OO1, which bears a very simple schema (two classes) and only three operations, to *OO7,* which is more generic and proposes a complex and tunable schema (ten classes), as well as fifteen complex operations. However, even *OO7,* the more elaborate of these benchmarks, is not generic enough to model other types of applications, such as financial, multimedia, or telecommunication applications (Tiwary, Narasayya, & Levy, 1995). Furthermore, its complexity makes it hard to understand and implement. To circumvent these limitations, the *OCB* benchmark has been proposed (Darmont & Schneider, 2000). Wholly tunable, this tool aims at being truly generic. Still, the benchmark's code is short, reasonably easy to implement, and easily portable. Finally, *OCB* has been extended into the Dynamic Evaluation Framework (DEF), which introduces a dynamic component in the workload, by simulating access pattern changes using configurable styles of changes (He & Darmont, 2005).

Object-relational benchmarks such as BUCKY (Carey, DeWitt, & Naughton, 1997) and BORD (Lee, Kim, & Kim, 2000) are query-oriented and solely dedicated to object-relational systems. For instance, BUCKY only proposes operations that are specific to these systems, considering that typical object navigation is already addressed by object-oriented benchmarks. Hence, these benchmarks focus on queries implying object identifiers, inheritance, joins, class and object references, multivalued attributes, query unnesting, object methods, and abstract data types.

XML Benchmarks

Since there is no standard model, the storage solutions for XML (eXtended Markup Language) documents that have been developed since the late nineties bear significant differences, both at the conceptual and the functionality levels. The need to compare these solutions, especially in terms of performance, has lead to the design of several benchmarks with diverse objectives.

X-Mach1 (Böhme & Rahm, 2001), XMark (Schmidt, Waas, Kersten, Carey, Manolescu, & Busse, 2002), XOO7 (an extension of OO7) (Bressan, Lee, Li, Lacroix, & Nambiar, 2002) and XBench (Yao, Ozsu, & Khandelwal, 2004) are so-called application benchmarks. Their objective is to evaluate the global performances of an XML DBMS, and more particularly of its query processor. Each of them implements a mixed XML database that is both data-oriented (structured data) and document-oriented (in general, random texts built from a dictionary). However, except for XBench that proposes a true mixed database, their orientation is more particularly focused on data (XMark, XOO7) or documents (X-Mach1).

These benchmarks also differ in:

- the fixed or flexible nature of the XML schema (one or several Document Type Definitions or XML schemas);
- the number of XML documents used to model the database at the physical level (one or several);
- the inclusion or not of update operations in the workload.

We can also underline that only XBench helps in evaluating all the functionalities offered by the XQuery language.

Micro-benchmarks have also been proposed to evaluate the individual performances of basic operations such as projections, selections, joins, and aggregations, rather than more complex queries. The Michigan Benchmark (Runapongsa, Patel, Jagadish, & Al-Khalifa, 2002) and MemBeR (Afanasiev, Manolescu, & Michiels, 2005) are made for XML documents storage solution designers, who can isolate critical issues to optimize, rather than for users seeking to compare different systems. Furthermore, MemBeR proposes a methodology for building micro-databases, to help users in adding datasets and specific queries to a given performance evaluation task.

Decision-Support Benchmarks

Since decision-support benchmarks are currently a *de facto* subclass of relational benchmarks, the TPC again plays a central role in their standardization. *TPC-H* (TPC, 2005c) is currently their only decision-support benchmark. It exploits a classical *product-order-supplier* database schema, as well as a workload that is constituted of twenty-two SQL-92, parameterized, decision-support queries, and two refreshing functions that insert tuples into, and delete tuples from, the database. Query parameters are randomly instantiated following a uniform law. Three primary metrics are used in *TPC-H*. They describe performance in terms of power, throughput, and a combination of these two criteria.

Data warehouses nowadays constitute a key decision-support technology. However, *TPC-H's* database schema is not a star-like schema that is typical in data warehouses. Furthermore, its workload does not include any On-Line Analytical Processing (OLAP) query. *TPC-DS*, which is currently under development (TPC, 2005b), fills in this gap. Its schema represents the decision-support functions of a retailer under the form of a constellation schema with several fact tables and shared dimensions. *TPC-DS*' workload is constituted of four classes of queries: reporting queries, *ad-hoc* decision-support queries, interactive OLAP queries, and extraction queries. SQL-99 query templates help in randomly generating a set of about five hundred queries, following non-uniform distributions. The warehouse maintenance process includes a full ETL (Extract, Transform, Load) phase, and handles dimensions according to their nature (non-static dimensions scale up while static dimensions are updated). One primary throughput metric is proposed in TPC-DS. It takes both query execution and the maintenance phase into account.

As in all the other *TPC benchmarks*, scaling in *TPC-H* and *TPC-DS* is achieved through a scale factor that helps defining the database's size (from 1 GB to 100 TB). Both the database schema and the workload are fixed.

There are, again, few decision-support benchmarks out of the *TPC*, and their specifications are rarely integrally published. Some are nonetheless interesting. APB-1 is presumably the most famous. Published by the OLAP Council, a now inactive organization founded by OLAP vendors, APB-1 has been intensively used in the late nineties. Its warehouse dimensional schema is structured around four dimensions: *Customer*, *Product*, *Channel*, and *Time*. Its workload of ten queries is aimed at sale forecasting. APB-1 is quite simple and proved limited to evaluate the specifici-