Evaluating the Dynamic Behavior of Database Applications

Zhen He, University of Vermont Burlington, USA
Jérôme Darmont, Université Lumière Lyon 2, France

ABSTRACT

This paper explores the effect that changing access patterns has on the performance of database management systems. Changes in access patterns play an important role in determining the efficiency of key performance optimization techniques, such as dynamic clustering, prefetching, and buffer replacement. However, all existing benchmarks or evaluation frameworks produce static access patterns in which objects are always accessed in the same order repeatedly. Hence, we have proposed the Dynamic Evaluation Framework (DEF) that simulates access pattern changes using configurable styles of change. DEF has been designed to be open and fully extensible (e.g., new access pattern change models can be added easily). In this paper, we instantiate DEF into the Dynamic Object Evaluation Framework (DoEF) which is designed for object databases, that is, object-oriented or object-relational databases, such as multimedia databases or most XML databases. The capabilities of DoEF have been evaluated by simulating the execution of four different dynamic clustering algorithms. The results confirm our analysis that flexible conservative reclustering is the key in determining a clustering algorithm’s ability to adapt to changes in access pattern. These results show the effectiveness of DoEF at determining the adaptability of each dynamic clustering algorithm to changes in access pattern in a simulation environment. In a second set of experiments, we have used DoEF to compare the performance of two real-life object stores: Platypus and SHORE. DoEF has helped to reveal the poor swapping performance of Platypus.

Keywords: benchmarking; clustering; dynamic access patterns; object-relational databases; performance evaluation

INTRODUCTION

Performance evaluation is critical for both designers of Database Management Systems (DBMSs) for architectural or optimization choices and users for efficiency comparison or tuning. Traditionally, this is achieved with the use of benchmarks, that is, synthetic workload models (databases and operations) and sets of performance metrics. Although in real life, almost no application always accesses the same data in the same order repeatedly, none of the existing database benchmarks incorporate the possibility of change in the access patterns. The ability to adapt to changes in access patterns is critical to database performance. In addition, highly tuning a data-
base to perform well for only one particular access pattern can lead to poor performance when different access patterns are used. Thus, a database tuned to a particular trace (a particular instance of a real application usage) is likely to perform poorly when a different trace is used. In addition, the performance of a database on a particular trace provides little insight into the reasons behind its performance and thus is of limited use to database researchers or engineers who are interested in the identification and improvement in the performance of particular components of the system.

Thus, the aim of our work is to provide a means for them to explore the performance of databases under different styles of access pattern change. In contrast, benchmarks of the TPC family aim to provide standardized means of comparing systems for vendors and customers. In this paper, we take a first look at how dynamic application behavior can be modeled and propose the Dynamic Evaluation Framework (DEF). DEF makes the first attempt at exploring the issue of evaluating the performance of DBMSs in general and such optimization techniques as dynamic clustering algorithms in particular with respect to changing query profiles. DEF contains a set of protocols which in turn define a set of styles of access pattern change. DEF by no means has exhausted all possible styles of access pattern change. However, we have designed DEF to be fully extensible, and its design allows new styles of change to be easily incorporated. Finally, DEF is a generic platform that can be specialized to suit the particular needs of a given family of DBMS (e.g., relational, object, or object-relational). In particular, it is designed to be implemented on top of an existing benchmark so that previous benchmarking research and standards can be reused.

In this paper, we show the utility of DEF by creating an instance of DEF called the Dynamic object Evaluation Framework (DoEF) (He & Darmont, 2003) which is designed for object databases.

Note that in the remainder of this paper, we term Object Database Management Systems (ODBMSs) both object-oriented and object-relational systems, indifferently. ODBMSs include most multimedia and XML DBMSs, for example. DoEF is built on top of the Object Clustering Benchmark (OCB) (Darmont, Petit & Schneider, 1998; Darmont & Schneider, 2000), which is a generic object-oriented benchmark that is able to simulate the behavior of other main object-oriented benchmarks. DoEF uses both the database built from the rich schema of OCB and the operations offered by OCB. Since OCB’s generic model can be implemented within an object-relational system and most of its operations are relevant for such a system, DoEF can also be used in the object-relational context.

To test the effectiveness of DoEF, we have conducted two sets of experiments. First, we have benchmarked four state-of-the-art dynamic clustering algorithms (Bullat & Schneider, 1996; Darmont et al., 2000; He, Marquez & Blackburn, 2000). There are three reasons for choosing to test the effectiveness of DoEF using dynamic clustering algorithms: (1) ever since the “early days” of object database management systems, clustering has been proven to be one of the most effective performance enhancement techniques (Gerlhof, Kemper & Moerkotte, 1996); (2) the performance of dynamic clustering algorithms is very sensitive to changing access patterns; and (3) despite this sensi-
Common Sense Reasoning in Automated Database Design: An Empirical Test
Veda C. Storey, Robert C. Goldstein and Jason Ding (2002). *Journal of Database Management* (pp. 3-14).
[www.igi-global.com/article/common-sense-reasoning-automated-database/3272?camid=4v1a](www.igi-global.com/article/common-sense-reasoning-automated-database/3272?camid=4v1a)

Integrating Digital Signatures with Relational Databases: Issues and Organizational Implications
[www.igi-global.com/article/integrating-digital-signatures-relational-databases/3294?camid=4v1a](www.igi-global.com/article/integrating-digital-signatures-relational-databases/3294?camid=4v1a)