Universidade do Minho
Escola de Engenharia

Semana da Escola de Engenharia
October 24 - 27, 2011

REPLICATION PROTOCOLS VS THE REAL WORLD

Ana Nunes and José Pereira
Department of Informatics
University of Minho
E-mail: {ananunes,jop}@di.uminho.pt

KEYWORDS
Database replication, Benchmarking

ABSTRACT

Several group communication-based database replication protocols have been proposed in recent years. Proposed protocols range from conservative to optimistic and while some require conflict classes to be disjoint, others only require conflict classes to be unique.

However, the tools commonly used to test these protocols offer a simplified model of reality. Also, the assumptions, regarding the flexibility of partitioning schemes of more complex applications, on which the database replication protocols are based seem to be too strong. We argue that this is not sufficient for suitably testing the applicability of these protocols in more realistic settings and provide some examples.

INTRODUCTION

Database replication has been a hot research topic for some time now, from single-tier architectures to multi-tier and cloud architectures. The focus has been on how to enable highly available applications/services through fault-tolerant and scalable architectures.

Group communication-based database replication protocols come in two flavors: conservative and optimistic. The distinction arises from the moment in which transactions are ordered with respect to transaction execution: in conservative protocols, the execution of potentially conflicting transactions takes places only after replicas decide on a total order; conversely, in optimistic protocols, replica coordination to detect conflicts among concurrent transactions is deferred to just before commit time.

Conflict classes are often used to implement concurrency control in database replication protocols. In short, the available data is partitioned according to some criteria, and a FIFO transaction queue is associated to each partition. Each transaction will then be enqueued in the queues of all conflict classes (data partitions) it accesses. The FIFO discipline is essential for preventing conflicts. Two transactions that access disjoint sets of basic conflict classes are guaranteed not to conflict and thus can be concurrently executed, while the probability of conflict among transactions that access the same class is high.

There are several replication protocols, both conservative and optimistic that are based on conflict classes. The manner in which the conflict classes are defined has a profound impact on the replication protocol. Some protocols require conflict classes to be disjoint (Kemme et al. 1999), while others (Jiménez-Peris et al. 2002, Patiño-Martínez et al. 2000) can handle non-disjoint conflict classes. Most of these protocols have been tested using straightforward benchmarks such as TPC-C (TPC 2001a) and TPC-W (TPC 2001b), for which partitioning schemes can be easily derived.

The question remains whether the assumptions made regarding conflict class definition remain plausible when dealing with more complex benchmarks, for which partitioning is not straightforward at all. The same question can be posed regarding real-world applications.

BACKGROUND

In (Kemme et al. 1999), the OTP replication protocol is based on basic conflict classes, which correspond to disjoint data partitions. Also, each transaction is restricted to accessing only one conflict class, which enables a passive replication scheme in a primary-backup configuration for each partition (class).

Compound conflict classes, defined as sets of basic conflict classes are introduced in (Jiménez-Peris et al. 2002, Patiño-Martínez et al. 2000) with the NODO protocol. Compound conflict classes do not need to be disjoint, but are required to be distinct, so that a primary-backup scheme can still be defined for each partition (class). However, transactions are still restricted to accessing only one conflict class.

With this primary-backup configuration, the conflict classes define the load distribution. However, no guarantee is given on the balance of that distribution. Notice that any flexibility in defining conflict classes is impaired by the restriction of each transaction accessing a single class.

In particular, the performance of conservative protocols hinges on a favorable definition of conflict classes, since the number of conflict classes defines the maximum number of concurrent transactions that can be executed.

The AKARA protocol (Correia et al. 2008) does not restrict transactions to accessing a single conflict class and is capable of doing both passive and active replication. Unlike NODO, AKARA does not rely on conflict classes to do load balancing. In the passive mode, the replica that executes the transaction is the one where it arrives. NODO and AKARA's performance was evaluated in (Correia et al. 2008) using a variation of the TPC-C benchmark.

DISCUSSION

The TPC-E benchmark (TPC 2010) simulates the activities of a brokerage firm which handles: customer account management, trade order execution on behalf of customers and interaction with financial markets. This benchmark defines 33 tables across four domains: customer, broker, market and dimension and 10 main transaction types that operate across the domains. Unlike TPC-C, TPC-E is an open benchmark suite, since new requests are received by the system under test regardless of the completion of previous requests.

This is a considerably more complex benchmark than TPC-C. While it is suggested in the benchmark specification that the customer domain can be straightforwardly partitioned by the customer identifier, that does not suffice for creating a conflict class definition suitable for yielding good performance from conservative replication protocols. Optimistic protocols such as AKARA which relax the admission control of transactions for execution, enabling the concurrent execution of transactions which could potentially conflict according to the conflict class criterion, will probably fare better in this type of system.

While the TPC-E benchmark is reasonably complex, most real-world applications are in all likelihood even more complex. Also, some, particularly in SME contexts are not as well-structured as TPC-E. We can thus reasonably assume that conclusions drawn from analyzing the benchmark in this aspect will likely be useful in understanding the behavior of database replication protocols in real-world applications.

REFERENCES

Correia, A.; J. Pereira; and R. Oliveira 2008. "Akara: A flexible clustering protocol for demanding transactional workloads." *On the Move to Meaningful Internet Systems: OTM 2008*, pages 691–708.

Jiménez-Peris, R.; M. Patiño-Martínez; B. Kemme; and G. Alonso. 2002. "Improving the scalability of fault-tolerant database clusters". In *Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on*, pages 477–484.

Kemme, B.; F. Pedone; G. Alonso; and A. Schiper.1999. "Processing transactions over optimistic atomic broadcast protocols". In *Distributed Computing Systems, 1999. Proceedings. 19th IEEE International Conference on*, pages 424–431.

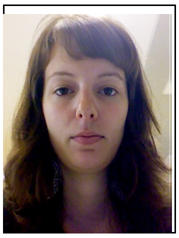
Patiño-Martínez, M; R. Jiménez-Peris; B. Kemme; and G. Alonso. 2000. "Scalable replication in database clusters". In *Distributed Computing*, pages 147–160.

Transaction Processing Performance Council(TPC). June 2010. *TPC Benchmark E - Standard Specification, Revision 1.12.0.*

Transaction Processing Performance Council(TPC). 2001. *TPC Benchmark C - Standard Specification, Revision 5.0.*

Transaction Processing Performance Council(TPC). August 2001. *TPC Benchmark W - Standard Specification, Revision 1.6.*

AUTHOR'S BIOGRAPHY



ANA NUNES was born in Vila Real, Portugal and attended the University of Minho, where she studied Informatics Engineering and obtained her degree in 2008. Later, she obtained her MSc degree in Informatics from the same university in 2009. She has since enrolled in the MAP-i Doctoral Program and is now pursuing a PhD in in the Distributed Systems area, under the specific theme of "Elastic Enterprise Applications". Her e-mail address is : ananunes@di.uminho.pt .