SPEED MEASUREMENTS OF DATABASE REPLICATION PREPARING FOR APPLICATION DEPENDENT PARTIAL REPLICATION OF DATABASES

Ágnes ERDŐSNÉ NÉMETH
Advisor: András PATARICZA

I. Introduction

With the increasingly widespread use of web-based services, databases constitute more and more critical parts of applications. System crashes involving the database lead to large recovery times, and accordingly to a low availability of the applications relying on it. The redundancy needed to assure the fault tolerance of large-scale databases, whether a complete logic replication of the database content or a RAID-like partial one (RAIDb) is used. The latter is in the scope of this article.

However, critical services use frequently only a minor fraction of the data in the database heavily and the larger part serves other purposes, e.g. logging. As the user perceived availability depends both on the frequency of use a particular service and its criticality and availability a cheap solution was searched to assure a high availability for the most critical services at the price of a moderate redundancy.

The aim of our work is to propose an application-dependent partial replication scheme assuring fast recovery of the most important data and consequently a high availability of the most critical services; while the recovery of less critical services is postponed until the main database becomes available again after a longer time.

For the composite measurements it is important to choose the type of the database server, a partial replication tool and a pilot application with critical web-based services.

II. Results of the analysis

RAIDb solutions offer a database independent solution in the terms of functionality, as this middleware communicates with the core database only via standard JDBC calls. However, the selection of a particular core database type may influence the performance of the RAIDb setup by reacting differently to the redundancy in the requests. Accordingly, at first, a small JAVA application was used to analyze different database types and replication methods:

- Implementations of the same database structure over two MySQL implementations offering different level of services (MyISAM offering only simplest database functions, and InnoDB supporting advanced features like atomic transactions as elementary operations) served as one group of baseline performance measurements. Additionally, IBM DB2 represented high-performance commercial database engines. 1000 and 10000 “insert/update” statements served as basic workload in this performance comparison experiment estimating the maximal performance by not involving any middleware or replication. The simple MySQL (MyISAM) and the commercial DB2 implementations had nearly the same response time, while the InnoDB solution was 20-35 times slower than the first two implementations.
- The next experiment aimed at the estimation of the performance overhead originating in a full replication scheme corresponding to a high level of fault tolerance executing an update of the replica immediately after a write operation to the database. This setup is quite similar to mirroring, but the performance test executes only write-type operations in order to avoid interferences from read speedup mechanisms. Measurements were performed both by using

the built-in full database replica mechanisms of the different database engines, and externally driven by Sequoia.[1][2]. Here once again MySQL with MyISAM type tables and IBM DB2 Q-Replication offer a similar performance both with the built-in and Sequoia-based replication, but the InnoDB is slower by a factor of 10-15 times. (See Figure 1.)

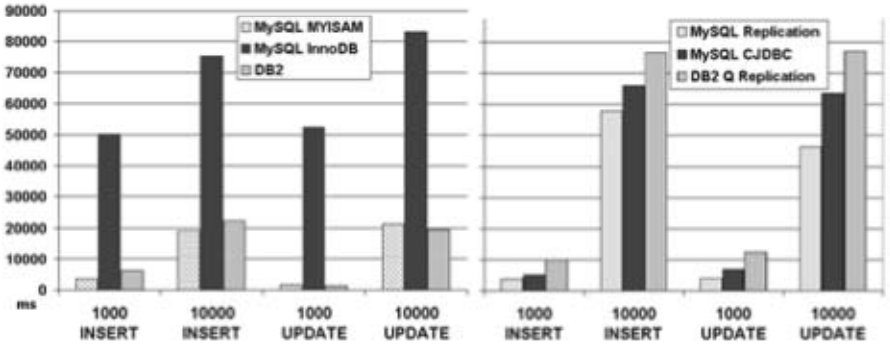

Figure 1: The measurements data

III. Conclusion

Partial replication is only available for IBM DB2 as a built-in feature. Q-Replication can execute a partial replication as fine granular, as the smallest unit of replication is a single attribute primarily used for database federation. The missing support in MySQL necessitates the use of Sequoia. Once again, the performance measures of MySQL for MyISAM and DB2 are quite similar.

The performance related experiences can be summarized in such a way, that simple databases offering a very basic set of services and a commercial one with sophisticated services may reach a similar performance for those basic operations, which are needed for replication.

IV. Future work

For the composite measurements, Sequoia with MySQL MyISAM tables and IBM DB2 Q-Replication were selected as database platform for partial replication (RAIDb-2) implemented with Sequoia and the TPC-W benchmark as pilot application (see figure 2.) [4] [5].

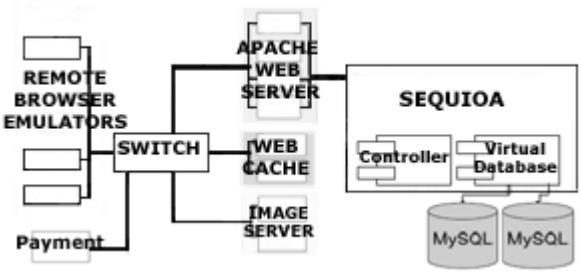

Figure 2: A The architecture of the TPC-W benchmark implementation with Sequoia

References

[1] E. Cecchet, *RAIDb: Redundant Array of Inexpensive Databases*, LNCS 3358, 2004, pp. 115 – 125, ISPA 2004

[2] E. Cecchet, J. Marguerite, W. Zwaenepoel, *Partial Replication: Achieving Scalability in Redundant Arrays of Inexpensive Databases*, LNCS 3144, 2004, pp 58 – 70, OPODIS 2003

[3] G. Pintér, H. Madeira, M. Vieira, I. Majzik, A. Pataricza, *A Data Mining Approach to Identify Key Factors in Dependability Experiments*, LNCS 3463, 2005, pp 263 – 280, EDCC 2005

[4] Specification of TPC-W: Transaction Processing Performance Council - http://www.tpc.org

[5] Implementation of TPC-W: http://www.ece.wisc.edu/~pharm/ & http://www.cs.cmu.edu/~manjhi/tpcw.html