Tutorial: Designing an Ultra Highly Available DBMS

Svein Erik Bratsberg
Clustra AS, 7485 Trondheim, Norway
svein.erik.bratsberg@clustra.com

Øystein Torbjørnsen
Clustra AS, 7485 Trondheim, Norway
oystein.torbjornsen@clustra.com

1 Introduction

Most database management systems available today are systems designed for general use. Certainly, some compromises have been done to satisfy the the most common users and the largest markets. One application which has been mostly ignored until now is the network equipment made for the telco operators.

The equipment used in the telco industry has requirements which have been quite different from the typical database applications, especially with respect to availability and real-time performance. This has caused the telco manufacturers to develop their own hardware, operating systems, programming languages and database management systems. There has been little standardization and cooperation causing a wide variety of solutions both between the vendors and within the vendors themselves. This has been possible because of high prices, long product development cycles and long product lifetimes.

Due to the deregulation of the market, increased competition, better price, quality and performance of standard products and an emerging Internet market with telco requirements, standard SW and HW products are becoming more and more fit for the telco market. Any database system trying to replace the proprietary solutions must satisfy the basic telco requirements: availability ($\geq$ 99.999%); real-time response (1-10 ms.); scalable throughput (10 to 100,000 TPS); open interfaces (SQL, ODBC, JDBC); run on commodity HW/SW.

For today's mainstream database management systems several of these requirements are not satisfied. Although claiming high availability, the systems can only achieve an availability which is one or two orders of magnitude worse than the requirements. The main obstacles for this are system maintenance and long takeover times in the case of failures. Another requirement not satisfied by mainstream systems is the response times for update transactions. They all synchronously write the log to disk before committing. Combined with a group commit strategy, transaction response times are highly variable and in the range of 50 milliseconds or higher for update transactions.

ACM SIGMOD 2000 5/00 Dallas, TX, USA

2 Outline

This tutorial will present the design issues for a DBMS solving the specific problems addressed above. It will run through the state-of-the-art and focus on the *Clustra Parallel Data Server*. We will present basic software and hardware architecture for a high availability DBMS, including interconnect and disk solutions.

Special focus will be made on fault-tolerance mechanisms, including replication, take-over, recovery and repair. We will also go through methods for on-line system maintenance, including on-line backup and restore, software and hardware upgrades and schema changes. Testing and performance issues are treated as well.

This tutorial is aimed for designers and developers planning to use a DBMS in the telco or Internet domain, and for everybody interested in DBMS internals in general.