Chapter 2.21
OLAP with a Database Cluster

Uwe Röhm
University of Sydney, Australia

ABSTRACT

This chapter presents a new approach to online decision support systems that is scalable, fast, and capable of analysing up-to-date data. It is based on a database cluster: a cluster of commercial off-the-shelf computers as hardware infrastructure and off-the-shelf database management systems as transactional storage managers. We focus on central architectural issues and on the performance implications of such a cluster-based decision support system. In the first half, we present a scalable infrastructure and discuss physical data design alternatives for cluster-based online decision support systems. In the second half of the chapter, we discuss query routing algorithms and freshness-aware scheduling. This protocol enables users to seamlessly decide how fresh the data analysed should be by allowing for different degrees of freshness of the online analytical processing (OLAP) nodes. In particular it becomes then possible to trade freshness of data for query performance.

INTRODUCTION

Online analytical processing (OLAP) systems must cope with huge volumes of data and at the same time must allow for short response times to facilitate interactive usage. They must also be capable to scale, meaning to be easily extensible with the increasing data volumes accumulated. Furthermore, the requirement that the data analysed should be up-to-date is becoming more and more important. However, not only are these contrary requirements, but they also run counter to the performance needs of the day-to-day business.

Most OLAP systems nowadays are kept separated from mission critical systems. This means that they offer a compromise between "up-to-dateness," that is, freshness (or currency) of data, and query response times. The data needed are propagated into the OLAP system on a regular basis, preferably when not slowing down day-to-day business, for example, during nights or weekends. OLAP users have no alternative but to analyse stale data.

But a decision support system that could provide decision makers insight into up-to-date data "hot off the press" would open exciting new possibilities. A stockbroker, for example, could analyse current trends in the market online. For e-commerce, the personalisation of Web shops could be much improved by more complex analysis of current browsing behaviour. Even for the so-called "old economy," new perspectives open up, because the update window has already become drastically small in a 24/7 setting of a worldwide operating company. However, up to now there is no solution that meets these performance and freshness requirements at the same time.

In this chapter, we present a new approach to online decision support systems that is capable of analysing up-to-date data. It is based on a database cluster: this is a cluster of commercial off-the-shelf computers as hardware infrastructure and off-the-shelf database management systems as transactional storage managers. A coordination middleware on top hides the details and provides a uniform, general-purpose query interface. The result is a "database of databases" following the vision of a hyperdatabase (Schek, Böhm, Grabs, Röhm, Schuldt, & Weber, 2000). An important design principle of a database cluster is its component-oriented nature. In particular, we want to be able to easily plug together and to expand the cluster using standard hardware and software components only. This results in a highly scalable system architecture.

We concentrate on central architectural issues and performance aspects of database clusters for usage in a decision support scenario. The objective is to develop a basic infrastructure for interactive decision support systems that are capable of analysing up-to-date data and that can give guarantees on how outdated data accessed might be. To be able to do so, we need different versions of data in the cluster, which we achieve by replicating data throughout the cluster. Replication also helps to avoid expensive distributed joins over huge amounts of data; as always several nodes can evaluate an OLAP query. We will discuss query routing strategies for an optimal workload distribution of long running and I/O intensive OLAP queries over encapsulated standard components of a database cluster.

Furthermore, we explicitly allow that not all cluster nodes are up-to-date all the time. This is reflected by the notion of freshness of data, which is a measure for the deviation of a certain component as compared to an up-to-date component. We present an innovative approach to replication management, called freshness-aware scheduling (FAS). The intention of freshness-aware scheduling is to trade query performance for freshness of data. Consequently, FAS introduces a new quality-of-service parameter that allows queries to specify an explicit freshness limit for the data accessed. If some queries agree to be evaluated on older data, update propagation can be interleaved with query processing more efficiently. This results in an overall better system performance and only a minimal slowdown of both queries and updates. In particular, it enables clients to request and access up-to-date data.

The remainder of this chapter is organised as follows: In the next two sections, we will first present a scalable infrastructure for an unified OLTP/OLAP database cluster and discuss physical data design alternatives for cluster-based online decision support systems. The subsequent two sections introduce query routing and freshness-aware scheduling, and also discuss related work. Both techniques have been prototypically implemented as part of the PowerDB project at ETH Zurich and in the Evaluation section, we report on the results of a comprehensive performance evaluation with our prototype system. The last section concludes the chapter.

Related Content