Chapter 8

ChunkSim
A Tool and Analysis of Performance and Availability Balancing

Pedro Furtado
University of Coimbra, Portugal

ABSTRACT

Self-tuning physical database organization involves tools that determine automatically the best solution concerning partitioning, placement, creation and tuning of auxiliary structures (e.g. indexes), based on the workload. To the best of our knowledge, no tool has focused on a relevant issue in parallel databases and in particular data warehouses running on common off-the-shelf hardware in a shared-nothing configuration: determining the adequate tradeoff for balancing load and availability with costs (storage and loading costs). In previous work, we argued that effective load and availability balancing over partitioned datasets can be obtained through chunk-wise placement and replication, together with on-demand processing. In this work, we propose ChunkSim, a simulator for system size planning, performance analysis against replication degree and availability analysis. We apply the tool to illustrate the kind of results that can be obtained by it. The whole discussion in the chapter provides very important insight into data allocation and query processing over shared-nothing data warehouses and how a good simulation analysis tool can be built to predict and analyze actual systems and intended deployments.

INTRODUCTION

Data warehouses may range from few megabytes to huge giga- or terabyte repositories, so they must have efficient physical database design and processing solutions to allow for efficient operation. Physical database design concerns the layout of data and auxiliary structures on the database server and, together with the execution engine and optimizer, has a large influence on the efficiency of the system. This is especially relevant in parallel architectures that are setup to handle huge data sets efficiently. The data sets are partitioned into nodes and processed in a parallel fashion in order to decrease the processing burden and, above all, to allow the system to return fast results for near-to-interactive data exploration.
For this reason, there has been a lot of emphasis on automatic workload-based determination of partitioning indexes, materialized views and cube configurations in the past.

We concentrate on a different issue of providing desired load and availability levels with minimum degree of replication in parallel partitioned architectures and on system planning, both in the presence of heterogeneity. Consider a shared-nothing environment created in an ad-hoc fashion, by putting together a number of PCs. A data warehouse can be setup at a low cost in such a context, but then not only partitioning but also heterogeneity and availability become relevant issues. These issues are dealt with efficiently using load- and availability- balancing, where data is partitioned into pieces and replicated into processing nodes; there is a controller node orchestrating balanced execution, and processing nodes ask the controller node for the next piece of the data to process whenever they are idle (on-demand processing). Under this scheme, the best possible performance and availability balancing is guaranteed if all nodes have all the data (fully mirrored data), but smaller degrees of replication (partial replication) also achieve satisfactory levels of performance and load balancing without the loading and storage burdens of full mirroring. The issue then is how to predict and take decisions concerning the degree of partial replication that is necessary and how to size a system, which are objectives of the ChunkSim simulator that we present in this work. There is a tradeoff between the costs of maintaining large amounts of replicas (loading and storage costs) and the efficiency of the system in dealing with both heterogeneity/non-dedication of nodes and availability limitations, and there is a need to size a system while taking heterogeneity and replication alternatives into consideration. ChunkSim is a what-if analysis tool for determining the benefit of performance-wise placement and replication degree for heterogeneity and availability balancing in partitioned, partially replicated on-demand processing. Our contribution is to propose the tool and the model underlying it and to use it for system planning and the analysis of placement and replication alternatives.

The chapter is organized as follows: in the Background section we review partitioning, replication and load-balancing. Our review of replication will include works on both low-level replication (Patterson et al. 1998), relation-wise replication (e.g. chained declustering by Hsiao et al. 1990) and OLAP-wise replication (e.g. the works by Akal et al. 2002, Furtado 2004 and Furtado 2005). Then we review basic query processing in our shared-nothing environment, in section 3. In section 4 we describe the ChunkSim Model and parameters, including also a discussion on Placement and Processing approaches. The ChunkSim tool and underlying model is discussed next, and finally we use the tool to analyze the merits of different placement and replication configurations.

**BACKGROUND**

Shared-nothing parallel systems (SN) are systems where a possibly large number of computers (nodes) are interconnected such that, other than the network, no other resources are shared. These architectures are scalable, in the sense that it is possible to add a large number of computing nodes to handle larger data sets efficiently. The idea is that, if each node is able to process its part independently of the remaining nodes, the system will have good scalability properties. On the other hand, interconnections between processing units may become a bottleneck if large amounts of data need to be exchanged between nodes. For this reason, physical database design and query processing optimizations are most relevant in SN systems. In this context, self-tuning database systems most often rely on what-if analysis to determine the relevant physical parameters and organization. For instance, the AutoAdmin project (Chaudhuri et al. 2007) supported the creation