

Chapter XXIV

Node Partitioned Data Warehouses: Experimental Evidence and Improvements

Pedro Furtado

University of Coimbra, Portugal

ABSTRACT

Data Warehouses (DWs) with large quantities of data present major performance and scalability challenges, and parallelism can be used for major performance improvement in such context. However, instead of costly specialized parallel hardware and interconnections, we focus on low-cost standard computing nodes, possibly in a non-dedicated local network. In this environment, special care must be taken with partitioning and processing. We use experimental evidence to analyze the shortcomings of a basic horizontal partitioning strategy designed for that environment, then propose and test improvements to allow

efficient placement for the low-cost Node Partitioned Data Warehouse. We show experimentally that extra overheads related to processing large replicated relations and repartitioning requirements between nodes can significantly degrade speedup performance for many query patterns. We analyze a simple, easy-to-apply partitioning and placement decision that achieves good performance improvement results. Our experiments and discussion provide important insight into partitioning and processing issues for data warehouses in shared-nothing environments.

INTRODUCTION

Data Warehouses (DWs) are repositories that typically store large amounts of data that have been extracted and integrated from transactional systems and various other operational sources. Those repositories are useful for online analytical processing (OLAP) and data mining analysis. Typical queries include both standard reporting and ad hoc analysis. They usually are complex and access very large volumes of data, performing time-consuming aggregations. Although data warehouses easily can reach many Giga or Terabytes, users still require fast answers to their analyses. Therefore, performance becomes a major concern in those systems. Although structures such as materialized views and specialized indexes improve response times for predicted queries, parallel processing can be used alone or in conjunction with those structures to offer a major performance boost and to guarantee speedup and scale-up, even for unpredicted ad hoc queries.

Parallel database systems are implemented using one of the following parallel architectures: shared-memory, shared-disk, shared nothing, hierarchical, NUMA (Valduriez & Ozsu, 1999). Each choice has implications for parallel query processing algorithms and data placement. In practice, parallel environments involve several extra overheads related to data and control exchanges between processing units and also concerning storage, so that all components of the system need to be designed to avoid bottlenecks that would compromise the whole processing efficiency. Some parts of the system even may have to account for the aggregate flow into/from all units. For instance, in shared-disk systems, the storage system, including controllers and connections to storage, have to be sufficiently fast in order to handle the aggregate of all accesses without becoming a significant bottleneck for I/O-bound applications. To handle potential bottlenecks, specialized, fast, and fully dedicated parallel hardware and interconnects are required. An at-

tractive alternative is to use a number of low-cost computer nodes in a shared-nothing environment, possibly in a non-dedicated local network, and design the system with special partitioning and processing care. In such an environment, each node has a basic database engine, and the system includes a middle layer providing parallelism to the whole environment. The Node Partitioned Data Warehouse (NPDW) is a generic architecture for partitioning and processing query-intensive data in such an environment. One of the objectives of the Node Partitioned Data Warehouse is to minimize the dependency on very fast, dedicated computing and data exchange infrastructures by optimizing partitioning and making use of replication whenever useful.

DeWitt and Gray (1992) review the major issues in parallel database systems implemented over conventional shared-nothing architectures. One of the major concerns when using such an architecture is to decide how to partition or to cluster relations into nodes, which raises the issue of how to determine the most appropriate partitioning and placement choice for a schema. Data warehouses are a specialized type of database with specific characteristics and requirements that may be useful in the partitioning and placement decision. They are mostly read-only, periodically loaded centralized repositories of data. Replication-related consistency issues are minor when compared to full-blown transactional systems.

The star schema (Kimball, 1996) is part of the typical data organization in a data warehouse, representing a multidimensional logic with a large central fact table and smaller dimension tables. Facts typically are very large relations with hundreds of gigabytes of historical details. Dimensions are smaller relations identifying entities by means of several descriptive properties.

In that context, a basic placement strategy for the simple star schema replicates dimensions and fully partitions the large central fact horizontally randomly. Figure 1 illustrates the simple placement strategy. The large fact F is partitioned

18 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/node-partitioned-data-warehouses/28568

Related Content

View Materialization in a Data Cube: Optimization Models and Heuristics

Vikas Agrawal, P. S. Sundararaghavan, Mesbah U. Ahmed and Udayan Nandkeolyar (2007). *Journal of Database Management* (pp. 1-20).

www.irma-international.org/article/view-materialization-data-cube/3372

Empirical Comparison of 3-D Virtual World and Face-to-Face Classroom for Higher Education

Xiaofeng Chen, Keng Siau and Fiona Fui-Hoon Nah (2012). *Journal of Database Management* (pp. 30-49).

www.irma-international.org/article/empirical-comparison-virtual-world-face/74702

Database Reverse Engineering

Jean-Luc Hainaut, Jean Henrard, Didier Roland, Jean-Marc Hick and Vincent Englebert (2009). *Handbook of Research on Innovations in Database Technologies and Applications: Current and Future Trends* (pp. 181-189).

www.irma-international.org/chapter/database-reverse-engineering/20702

Research on Reverse Skyline Query Algorithm Based on Decision Set

Lan Huang, Yuanwei Zhao, Pedro Mestre, Laipeng Han, Kangping Wang, Wenjuan Gao and Rui Zhang (2022). *Journal of Database Management* (pp. 1-28).

www.irma-international.org/article/research-on-reverse-skyline-query-algorithm-based-on-decision-set/313971

Management Information Systems Needs for Public Service Delivery in the Digital Era

(2019). *Information Systems Strategic Planning for Public Service Delivery in the Digital Era* (pp. 166-226).

www.irma-international.org/chapter/management-information-systems-needs-for-public-service-delivery-in-the-digital-era/233408