Improved Data Partitioning for Building Large ROLAP Data Cubes in Parallel

Ying Chen, Dalhousie University, Canada
Frank Dehne, Carleton University, Canada
Todd Eavis, Concordia University, Canada
A. Rau-Chaplin, Dalhousie University, Canada

ABSTRACT

This paper presents an improved parallel method for generating ROLAP data cubes on a shared-nothing multiprocessor based on a novel optimized data partitioning technique. Since no shared disk is required, our method can be used for highly scalable processor clusters consisting of standard PCs with local disks only, connected via a data switch. Experiments show that our improved parallel method provides optimal, linear, speedup for at least 32 processors. The approach taken, which uses a ROLAP representation of the data cube, is well suited for large data warehouses and high dimensional data, and supports the generation of both fully materialized and partially materialized data cubes.

Keywords: data cube; parallel computing; ROLAP

INTRODUCTION

The precomputation of the different views (group-bys) of a data cube (i.e., the forming of aggregates for every combination of GROUP-BY attributes, is critical to improving the response time of Online Analytical Processing (OLAP) queries in decision support systems and can be instrumental in accelerating data mining tasks in large data warehouses (Han et al., 1996). As the size of data warehouses grows, the time it takes to perform this precomputation becomes a significant performance bottleneck, one which may stretch into days in the very largest cases (Microsoft, 2001). This article presents
an improved parallel method for generating ROLAP data cubes on shared-nothing multiprocessors, based on a novel optimized data partitioning technique. Since no shared disk is required, this method can be applied to highly scalable processor clusters consisting of standard PCs with local disks, connected via a high bandwidth (Ethernet) switch. Parallelism based on such shared-nothing machines is an attractive solution to improving system performance, especially in the context of large data warehouses where scaling I/O bandwidth to disk is as important as scaling computational resources.

For a given raw data set, \( R \), with \( N \) records and \( d \) attributes (dimensions), a view is constructed by an aggregation of \( R \) along a subset of attributes. As proposed in Gray et al. (1997), the precomputation of the full data cube (the set of all \( 2^d \) possible views) or a partial data cube (a subset of all \( 2^d \) possible views) supports the fast execution of subsequent OLAP queries. Many methods have been presented for generating the data cube on sequential (Beyer & Ramakrishnan, 1999; Harinarayan et al., 1996; Ross & Srivastava, 1997; Sarawagi et al., 1996; Yu & Lu, 2001; Zhao et al., 1997) and parallel systems (Chen et al., 2004; Dehne et al., 2001; Dehne et al., 2002; Goil & Choudhary, 1997; Goil & Choudhary, 1999; Lu et al., 1997; Muto & Kitsuregawa, 1999; Ng et al., 2001). For parallel data cube construction, good data partitioning is a key factor in obtaining good performance on shared nothing multiprocessors. Some researchers partition data on one or several dimensions (Goil & Choudhary, 1998; Muto & Kitsuregawa, 1999). They assume that the product of the cardinalities of these dimensions is much larger than the number of processors (Goil & Choudhary, 1998) in order to achieve sufficient parallelism. The advantage of their method is that they do not need to merge views across the network. However, in practice, this assumption is often not true. The cardinality of some dimensions may be small, such as gender, months, and intervals, for a numeric attribute. Therefore, those methods often are not scalable. One approach that avoids these problems is to partition on all dimensions and then to apply a parallel merge procedure (Chen et al., 2004). The challenge here is that merge procedures based on fixed data partitioning schemes often lead to excess interprocessor communications, which may greatly reduce the speedup achieved by the parallel system and limit its effective scalability.

In this article, we describe and evaluate a novel optimized data partition scheme for parallel ROLAP data cube generation, which improves significantly on our previous result in Chen et al. (2004) and outperforms all previously reported parallel data cube generation methods. Our new dynamic data partitioning scheme adapts to both the current data set and the performance parameters of the parallel machine. Using this scheme, data cube generation tasks involving millions of rows of input, which take days to perform on a single processor machine, can be completed in just hours on a 32-processor cluster. We have performed an extensive performance evaluation of our new method,
Related Content

Efficient and Effective Aggregate Keyword Search on Relational Databases
Luping Li, Stephen Petschulat, Guanting Tang, Jian Pei and Wo-Shun Luk (2012). *International Journal of Data Warehousing and Mining* (pp. 41-81).
[www.igi-global.com/article/efficient-effective-aggregate-keyword-search/74755?camid=4v1a](www.igi-global.com/article/efficient-effective-aggregate-keyword-search/74755?camid=4v1a)

Optimization of a Hybrid Methodology (CRISP-DM)
[www.igi-global.com/chapter/optimization-hybrid-methodology-crisp/73532?camid=4v1a](www.igi-global.com/chapter/optimization-hybrid-methodology-crisp/73532?camid=4v1a)

Finding the Semantic Relationship Between Wikipedia Articles Based on a Useful Entry Relationship
[www.igi-global.com/article/finding-the-semantic-relationship-between-wikipedia-articles-based-on-a-useful-entry-relationship/188489?camid=4v1a](www.igi-global.com/article/finding-the-semantic-relationship-between-wikipedia-articles-based-on-a-useful-entry-relationship/188489?camid=4v1a)

TtoO: Mining a Thesaurus and Texts to Build and Update a Domain Ontology
[www.igi-global.com/chapter/ttoo-mining-thesaurus-texts-build/7575?camid=4v1a](www.igi-global.com/chapter/ttoo-mining-thesaurus-texts-build/7575?camid=4v1a)