Chapter 5
Dynamic View Selection for OLAP

Michael Lawrence
Dalhousie University, Canada

Andrew Rau-Chaplin
Dalhousie University, Canada

ABSTRACT

In a data warehousing environment, aggregate views are often materialized in order to speed up aggregate queries of online analytical processing (OLAP). Due to the increasing size of data warehouses, it is often infeasible to materialize all views. View selection, the task of selecting a subset of views to materialize based on updates and expectations of the query load, is an important and challenging problem. In this article, we explore dynamic view selection in which the distribution of queries changes over time and the set of materialized views must be tuned by replacing some of the previously materialized views with new ones.

INTRODUCTION

An online analytical processing (OLAP) data warehouse answers queries interactively posed by users, the answers to which are used to support data-driven decision making. Such queries usually make heavy use of aggregation, which may be realized using the group by clause in SQL. Since aggregate queries are so common in OLAP environments and their results are typically very expensive to compute, aggregate views of the data are often pre-computed and stored in order to speed up future query processing. From the perspective of efficient query answering, ideally all views would be pre-computed and made available for answering aggregate queries. Realistically however, storage and computational constraints limit the number of views that are feasibly pre-materialized.

The problem of choosing a set of views for materialization is known as the View Selection Problem. In the view selection problem, one
wishes to select a set of views for materialization which minimizes one or more objectives, possibly subject to one or more constraints. Many variants of the view selection problem have been studied including minimizing the query cost of a materialized view set subject to a storage size constraint (Harinarayan, Rajaraman & Ullman, 1996; Shukla, Deshpande & Naughton 1998; Chirkova, Halevy & Suciu, 2002; Kalnis, Mamoulis & Papadias, 2002; Nadeau & Teorey, 2002), minimizing a linear combination of query cost and maintenance cost of a materialized view set (Barralis, Paraboschi, & Teniente, 1997; Gupta, 1997; Gupta, Harinarayan, Rajaraman, & Ullman, 1997; Theodoratos & Sellis, 1997, 1999, 2000; Yang, Karlapalem, & Li, 1997a, 1997b; Horng, Chang, Liu, & Kao, 1999; Uchiyama, Runapongs, & Teorey, 1999; Theodoratos, Dalmagas, Simitis, & Stavropoulos, 2001; Theodoratos, Ligoudistianos, & Sellis, 2001; Zhang, Yao & Yang, 2001; Valluri, Vadapalli, & Karlapalem, 2002; Gupta & Mumick, 2005), minimizing both query and maintenance cost as separate objectives (Lawrence, 2006), and minimizing query cost under a maintenance cost constraint (Gupta & Mumick, 1999; Lee & Hammer, 2001; Liang, Wang, & Orlowska, 2001; Kalnis et al., 2002; Yu, Yao, Choi, & Gou, 2003).

Note, however, that in most of these cases the problem considered is static in that:

1. Query frequencies are assumed to be static (i.e., not changing over time)
2. It is assumed that the pool of materialized views is to be selected and computed from scratch rather than making use of a running OLAP system’s pool of previously materialized views.

While the static view selection problem is important, it captures only the start-up phase of a OLAP system and does not address what is arguably in practice the more important dynamic question: how, given a running OLAP system with an existing pool of materialized views and a new vector of query frequencies, should we identify views that should be added to our materialized pool and views that should be removed in order to best minimize query times subject to a storage size constraint?

The need for dynamic view management was forcefully made by Kotidis and Roussopoulos (1999, 2001) in their DynaMat system. As they observe, “This static selection of views [...] contradicts the dynamic nature of decision support analysis.” There are a number of ways to approach the dynamic view selection problem. Kotidis and Roussopoulos (1999, 2001) take a caching approach in which rather than a set of materialized views at the data warehouse, a pool of view fragments is maintained. A view fragment is a portion of a whole view which results from a range selection on its dimensions. The authors describe how to locate fragments to answer a query, and how to decide which fragments to admit to the cache based on bounds on the size or maintenance cost of the cached fragments. Another approach is to consider the view selection problem in the context of a given set of queries (Gupta, 1997; Yang et al., 1997a, 1997b; Gupta & Mumick, 1999, 2005; Horng et al., 1999; Zhang et al., 2001; Valluri et al., 2002). The idea is to construct a directed acyclic graph (DAG) representing a joint query processing plan and then to consider the view selection problem consisting of choosing a subset of the nodes in the plan DAG so query cost is minimized under some set of constraints.

In this article, we explore an alternative approach to dynamic view selection. We consider an OLAP system with two phases of operation: Startup and Online. In the Startup Phase, an initial set of views must be selected based on some estimated query probabilities. This is the classical (static) view selection problem. In the Online Phase, an “in use” OLAP system is considered, for which a set of views $M$ has already been selected and materialized. Since over time the relative importance of each type of aggregate query may change due to the changing demands
Related Content

Role of Social Networking Sites in Enhancing Teaching Environment
[www.igi-global.com/chapter/role-of-social-networking-sites-in-enhancing-teaching-environment/178376?camid=4v1a](www.igi-global.com/chapter/role-of-social-networking-sites-in-enhancing-teaching-environment/178376?camid=4v1a)

Metaphors and Models for Data Mining Ethics
[www.igi-global.com/chapter/metaphors-models-data-mining-ethics/29143?camid=4v1a](www.igi-global.com/chapter/metaphors-models-data-mining-ethics/29143?camid=4v1a)

An Efficient Method for Discretizing Continuous Attributes
[www.igi-global.com/article/efficient-method-discretizing-continuous-attributes/42149?camid=4v1a](www.igi-global.com/article/efficient-method-discretizing-continuous-attributes/42149?camid=4v1a)

SpyNetMiner: An Outlier Analysis to Tag Elites in Clandestine Social Networks
[www.igi-global.com/article/spynetminer/106861?camid=4v1a](www.igi-global.com/article/spynetminer/106861?camid=4v1a)