Yet Another Automated OLAP Workload Analyzer: Principles, and Experiences

Alfredo Cuzzocrea\textsuperscript{1,2}, Rim Moussa\textsuperscript{3} and Enzo Mumolo\textsuperscript{1}

\textsuperscript{1}DIA Department, University of Trieste, Italy
\textsuperscript{2}ICAR-CNR, Italy
\textsuperscript{3}LaTICE and University of Carthage, Tunisia

Keywords: Data Warehouse Tuning, OLAP Intelligence, Data Warehouse Workloads, OLAP Workloads.

Abstract: In order to tune a data warehouse workload, we need automated recommenders on when and how (i) to partition data and (ii) to deploy summary structures such as derived attributes, aggregate tables, and (iii) to build OLAP indexes. In this paper, we share our experience of implementation of an OLAP workload analyzer, which exhaustively enumerates all materialized views, indexes and fragmentation schemas candidates. As a case of study, we consider TPC-DS benchmark -the de-facto industry standard benchmark for measuring the performance of decision support solutions including.

1 INTRODUCTION

Decision Support Systems (DSS) are designed to empower the user with the ability to make effective decisions regarding both the current and future activities of an organization. One of the most prominent technologies for knowledge discovery in DSS environments are On-line Analytical Processing (OLAP) technologies. OLAP relies heavily upon a data model known as the multidimensional database and the Data cube. The latter has been playing an essential role in the implementation of OLAP (Gray et al., 1997; Vassiliadis, 1998a). However, challenges related to Performance Tuning are to be addressed. OLAP workload Performance Tuning is usually based on (i) indexes, (ii) summary data, i.e. derived attributes and aggregate tables, and (iii) data fragmentation.

The paper outline is the following, in Section II, we overview Performance Tuning Strategies, from developer perspective. In Section III, we present our workload analyzer and our first experience with TPC-DS Benchmark. Finally we conclude the paper.

2 OLAP WORKLOAD PERFORMANCE TUNING

The term On-line Analytical Processing (OLAP) is introduced in 1993 by E. Codd (Codd et al., 1993). This model constitutes a decision support system framework which affords the ability to calculate, consolidate, view, and analyze data according to multiple dimensions. OLAP relies heavily upon a data model known as the multidimensional databases (MDB) (Kimball and Ross, 2013; Kimball et al., 1998; Molina, 2013; Imhoff et al., 2003; Inmon, 2005; DeWitt et al., 2005; Surajit and Umeshwar, 1997; Codd et al., 1993; Agarwal et al., 1996; Gyssens and Lakshmanan, 1997; Agrawal et al., 1997; Gray et al., 1997; Vassiliadis, 1998a). An MDB schema contains a logical model consisting of OLAP cubes. Each OLAP Cube is described by a fact table (facts), a set of dimensions and a set of measures. Multiple MDB design methods were proposed in the literature and are described in (Vassiliadis, 1998b; Cabibbo and Torello, 1998; Niemi et al., 2001; Hung et al., 2004; Nair et al., 2007; Malinowski and Zimányi, 2008; Romero and Abelló, 2009; Thanisch et al., 2011). In (Cuzzocrea and Moussa, 2013; Cuzzocrea et al., 2013a), we detail a framework for MDB schemas design, successfully applied to turn TPC-H benchmark into a multi-dimensional benchmark TPC-H*\textsuperscript{d}. In order to tune a data warehouse workload, we need automated recommenders on when and on how (i) to partition data and (ii) to deploy summary structures (e.g. derived attributes, aggregate tables, sketches synopsis, histograms synopsis), and (iii) to build OLAP indexes.
Many research work investigated distributed relational data warehouses and an adjunct mid-tier for parallel cube calculus, namely OLAP* (Cuzzocrea et al., 2013b), Other are investigating new systems SQL-on-Hadoop Systems (e.g., Apache Hive, Apache Spark, SQL, Apache Drill, Cloudera Impala, IBM BigInsights). Partitioning schemes are very important, good data fragmentation schemes allows parallel IO and parallel processing. Automated distributed database design was investigated in many research papers and by DBMS vendor leaders AutoPart (Papadomanolakis and Ailamaki, 2004), DB2 Design Advisor (Zilio et al., 2004), Database Tuning Advisor for MS SQL Server (Agrawal et al., 2004a; Agrawal et al., 2004b), and DDB-Expert (Moussa, 2011).

Indexes and Materialized Views are physical structures which aim at accelerating performance, like similarly OLAP query approximation approaches (e.g., (Cuzzocrea et al., 2009; Cuzzocrea and Matrangolo, 2004)). Many research papers cover automated selection of materialized views and indexes for OLAP workloads AutoAdmin (Agrawal et al., 2006), Alerters Approach (Hose et al., 2008), Semi-Automatic Index Tuning (Schnaitter and Polyzotis, 2012), AutoDB (Cuzzocrea and Moussa, 2013; Cuzzocrea et al., 2013a). Related work report experiences with TPC-H benchmark (Transaction Processing Council, 2013b). The latter is obsolete now. Its successor TPC-DS (Transaction Processing Council, 2013a) is the de facto industry standard benchmark for measuring the performance of decision support solutions. In this paper, we turn TPC-DS into a multidimensional benchmark and we analyze TPC-DS benchmark.

3 A MULTI-DIMENSIONAL DATABASE TPC-DS

There are few decision-support benchmarks out of the TPC benchmarks. Next, we overview most known DSS benchmarks, APB-1 (OLAP Council) has been released in 1998 by the OLAP council. APB-1 warehouse dimensional schema is structured around five fixed size dimensions and its workload is composed of 10 queries. APB-1 is proved limited (Erik, 1998) to evaluate the specificities of various activities. It proposes a single performance metric termed AQM (Analytical Queries per Minute). The metric AQM denotes the number of analytical queries processed per minute including data loading and computation time.

The most prominent benchmarks for evaluating decision support systems are the various benchmarks issued by the Transaction Processing Council (TPC).

Since two decades, TPC-H benchmark is the most used benchmark in the research community. The TPC-H benchmark (Transaction Processing Council, 2013b) exploits a classical product-order-supplier model. It consists of a suite of business oriented ad-hoc queries and concurrent data modifications. The workload is composed of twenty-two parameterized decision-support SQL queries with a high degree of complexity and two refresh functions: RF-1 new sales (new inserts) and RF-2 old sales (deletes). The TPC-DS benchmark is launched for next generation of decision support system benchmarking to replace the TPC-H benchmark. It is described in next Section.

3.1 TPC-DS Benchmark

TPC-DS (Transaction Processing Council, 2013a) was designed to examine large volumes of data, execute complex queries of various operational requirements and complexities (e.g., ad-hoc, reporting, iterative OLAP, data mining) within large number of user sessions. The benchmark stresses hardware system performance in the areas of CPU utilization, memory utilization, I/O subsystem utilization, and the ability of the operating system and database software to perform TPC-DS workload. The TPC-DS schema models seven data marts the sales and sales returns process for an organization that employs three primary sales channels: store, catalogs, and the Internet, as well as the Inventory. All data is periodically synchronized with source OLTP databases through database maintenance functions. The schema includes 7 fact tables and 17 dimension tables.

- Fact tables: store_sales, store_returns, catalog_sales, catalog_returns, web_sales, web_returns, inventory.
- Dimension tables: store, call_center, catalog_page, web_site, web_page, warehouse, customer, customer_address, customer_demographics, date_dim, household_demographics, item, income_band, promotion, reason, ship_mode, time_dim.

TPC-DS workload contains 99 SQL queries, covering SQL99, SQL-2003 (Eisenberg et al., 2004) (i.e., window functions) and OLAP capabilities. TPC-DS benchmark reports two main metrics (i) the Query-per-Hour Performance Metric ($/Qph) and (ii) The Price-Performance Metric ($/Qph) which reflects the ratio of costs to performance.
3.2 Turning TPC-DS Benchmark into a Multi-dimensional Benchmark

In order to turn the TPC-DS benchmark into a multidimensional benchmark, an initial schema is formed. The initial schema consists of all the cubes required to efficiently answer the TPC-DS queries. Each query is mapped to a minimal number of OLAP cubes. We design each OLAP cube with the relevant fact table, dimensions and measures. This leads to the definition of multiple cubes. Hereafter, we detail the process leading to the definition of each cube. We used the framework for automating multidimensional database schema design detailed in (Cuzzocrea and Moussa, 2013; Cuzzocrea et al., 2013a).

OLAP hypercube Cube 91 shown in Figure 1 is defined as a transform of Q91 (illustrated in Figure 2) into an OLAP hypercube. In the example, Cube 91 is an OLAP cube for Q91 of TPC-DS Benchmark (Transaction Processing Council, 2013a). Cube 91 has six dimensions (i) ‘Call Center’, (ii) ‘Returned Date’, (iii) ‘Returning Customer Marital Status’, (iv) ‘Returning Customer Education Status’, (v) ‘Returning Customer GMT Offset’ and (vi) ‘Buy Potential’ and one numeric measure ‘Sum of all Returns’ Net Losses’, and performs over ‘Catalog Returns facts’.

4 OLAP WORKLOAD ANALYZER

Tuning a database is a process that includes selection of indexes, materialized views, derived attributes, and fragmentation schemas. There are a number of tools that have been designed to take the responsibility from the database designer to advise the designer on good choices; SAP, Oracle, Vertica, PoWA of postgres, Teradata.

4.1 TPC-DS Numbers

We parse cubes (XML files), detect common dimensions and measures as well as different dimensions and measures for each pair of cubes.

4.2 Candidates Enumeration

The tuning advisor generates candidate indexes, materialized views, derived attributes, and fragmentation schemas and assesses the weight of each recommendation based on one or combination of these recommendations. We implemented a greedy approach to choosing indexes, materialized views, derived attributes and fragmentation schemas. Indeed, we enumerate automatically all candidate indexes, materialized views, derived attributes and fragmentation schemas.

- **Candidate Indexes**: For each cube, we consider indexes on foreign keys for the fact table, or join
indexes, simple and composite indexes attributes of dimension tables. For each dimension table with \( n \) attributes invoked for the calculus of cube, the number of indexes is \( \binom{n}{1} + \binom{n}{2} + \binom{n}{3} + \ldots + \binom{n}{n} \). Indexes types depend on cardinality of the dimension. Indeed, bitmaps are proposed for low cardinality dimensions and B-Tree based indexes are proposed for high cardinality dimensions. In practice, this choice is one of the principal factors that influence whether a database design gives acceptable performance. Two important factors to consider are: (i) The existence of an index on an attribute may speed up greatly the execution of those queries in which a value, or range of values, is specified for that attribute, and may speed up joins involving that attribute as well; (ii) On the other hand, every index built for one or more attributes of some relation makes insertions, deletions, and updates to that relation more complex and time-consuming.

- **Candidate Materialized Views**: For each a \( n \) dimensional cube, Based on the ALL values, the data cube is divided into \( 2^n \) cuboids. A materialized view is proposed for each cuboid. For instance, for Cube91, the first cuboid -the core cuboid- is a six dimensional cube (hexeract). The next \( \binom{6}{3} \) cuboids are five-dimensional cuboids. The next \( \binom{6}{2} \) are four-dimensional cuboids. The last cuboid has a single value and is a zero-dimensional point.

- **Candidate Derived Attributes**: For each cube, we check high cardinality snowflake dimensions (i.e., dimensions which cardinality is scale factor), and propose derived attributes within star dimensions (i.e., connecting through hierarchical relationships snowflake dimensions to the fact table). Derived attributes sketch all required measures.

- **Candidate Fragmentation Schemas**: We refer to OLAP* framework for generating candidate schema candidates.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we derived from TPC-DS benchmark a multi-dimensional database and reported a thorough analysis of TPC-DS benchmark, as well as the recommendations derived from the workload analysis. Each recommendation is characterized by a building cost estimation, a maintenance cost, a storage cost, and a weight in the workload. In Future work, we will investigate relationships among recommendations, i.e., namely consolidation and conflict relationships, in order to prune candidates combinations, and assess experimentally cubes calculus performances.

REFERENCES


Yet Another Automated OLAP Workload Analyzer: Principles, and Experiences

is it time to rethink the aggregate configuration of your


Data Warehouse Design: Relational and Dimensional Techniques. Wiley.


Kimball, R., Reeves, L., Thornha, W., Ross, M., and


