Yet Another Automated OLAP Workload Analyzer: Principles, and Experiences

Alfredo Cuzzocrea[1,2], Rim Moussa[3] and Enzo Mumolo[1]
[1]*DIA Department, University of Trieste, Italy*
[2]*ICAR-CNR, Italy*
[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 developper 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 litterature and are described in (Vassiliadis, 1998b; Cabibbo and Torlone, 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*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.

Cuzzocrea, A., Moussa, R. and Mumolo, E.
Yet Another Automated OLAP Workload Analyzer: Principles, and Experiences.
In *Proceedings of the 20th International Conference on Enterprise Information Systems (ICEIS 2018) - Volume 1*, pages 293-298
ISBN: 978-989-758-298-1

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), *Alerter Approach* (Hose et al., 2008), *Semi-Automatic Index Tuning* (Schnaitter and Polyzotis, 2012), *AutoMDB* (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 defacto 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 adhoc 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@Size* and *(ii)* The *Price-Performance Metric* ($/Qph) which reflects the ratio of costs to performance.

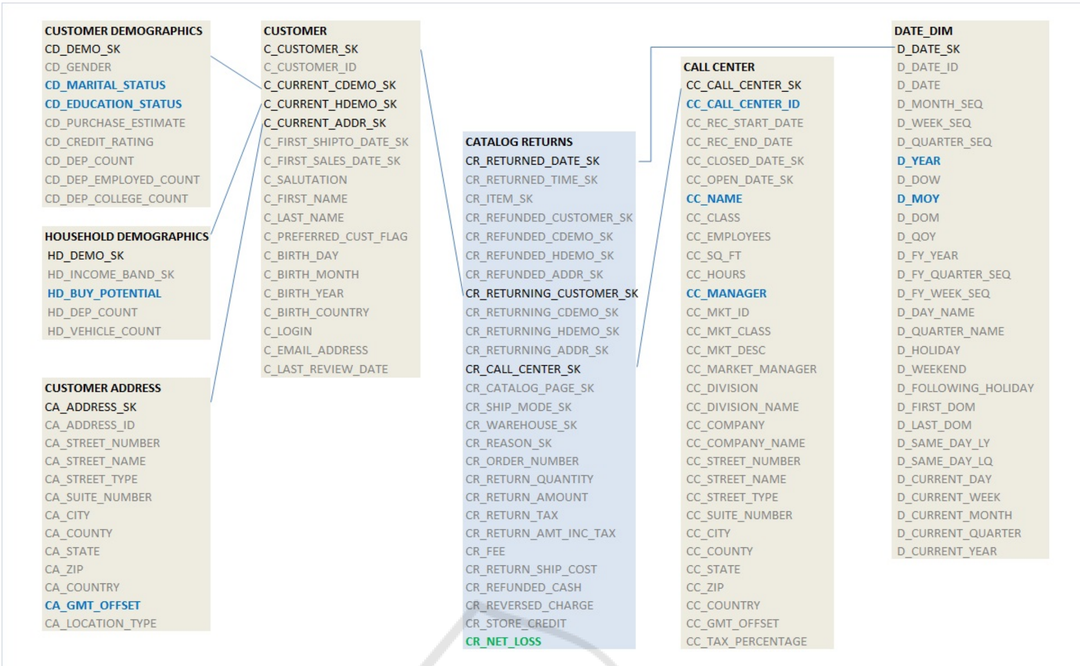

Figure 1: Data View of *TPC-DS Cube 91* -a sub-view of *Catalog Returns Datamart*.

3.2 Turning TPC-DS Benchmark into a Multi-dimensional Benchmark

In order to turn the TPC-DS benchmark into a mulmtidimensional 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, fragmentation schemas and assesses the weight of each recommendation based 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

```
Define YEAR = random(1998,2002, uniform);
Define MONTH = random(11,12,uniform);
Define BUY_POTENTIAL = text({"1001-5000",1},
{">10000",1},{"501-1000",1},{"0-500",1},
{"Unknown",1},{"5001-10000",1});
Define GMT = text({"-6",1},{"-7",1});

SELECT  cc_call_center_id Call_Center,
        cc_name Call_Center_Name,
        cc_manager Manager,
        SUM(cr_net_loss) Returns_Loss
FROM call_center,
        catalog_returns,
        date_dim,
        customer,
        customer_address,
        customer_demographics,
        household_demographics
WHERE cr_call_center_sk = cc_call_center_sk
AND cr_returned_date_sk = d_date_sk
AND cr_returning_customer_sk = c_customer_sk
AND cd_demo_sk = c_current_cdemo_sk
AND hd_demo_sk = c_current_hdemo_sk
AND ca_address_sk = c_current_addr_sk
AND d_year = [YEAR]
AND d_moy = [MONTH]
AND ( (cd_marital_status = 'M' AND
  cd_education_status = 'Unknown')
  OR (cd_marital_status = 'W' AND
  cd_education_status = 'Advanced Degree'))
AND hd_buy_potential like '[BUY_POTENTIAL]%'
AND ca_gmt_offset = [GMT]
GROUP BY cc_call_center_id, cc_name, cc_manager,
       cd_marital_status, cd_education_status
ORDER BY SUM(cr_net_loss) DESC;
```

Figure 2: SQL Statement of *TPC-DS* Query *Q91*.

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 $\binom{n}{1} + \binom{n}{2} + \binom{n}{3} + ... + \binom{n}{n}$. 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 $\binom{6}{5}$ cuboids are five-dimensional cuboids. The next $\binom{6}{4}$ 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 cardinaly 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

Agarwal, S., Agrawal, R., Deshpande, P., Gupta, A., Naughton, J. F., Ramakrishnan, R., and Sarawagi, S. (1996). On the computation of multidimensional aggregates. In *Proceedings of the 22th International Conference on Very Large Data Bases*, VLDB '96, pages 506–521. Morgan Kaufmann Publishers Inc.

Agrawal, R., Gupta, A., and Sarawagi, S. (1997). Modeling multidimensional databases. In *Proceedings of the Thirteenth International Conference on Data Engineering*, pages 232–243.

Agrawal, S., Bruno, N., Chaudhuri, S., and Narasayya, V. R. (2006). Autoadmin: Self-tuning database systemstechnology. *IEEE Data Eng. Bull.*, 29(3):7–15.

Agrawal, S., Chaudhuri, S., Kollár, L., Marathe, A. P., Narasayya, V. R., and Syamala, M. (2004a). Database tuning advisor for microsoft SQL server 2005. In *(e)Proceedings of the Thirtieth International Conference on Very Large Data Bases*, pages 1110–1121.

Agrawal, S., Narasayya, V. R., and Yang, B. (2004b). Integrating vertical and horizontal partitioning into automated physical database design. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Paris, France, June 13-18, 2004*, pages 359–370.

Cabibbo, L. and Torlone, R. (1998). A logical approach to multidimensional databases. In *Advances in Database Technology - EDBT'98, 6th International Conference on Extending Database Technology Proceedings*, pages 183–197.

Codd, E. F., Codd, S. B., and Salley, C. T. (1993). Providing OLAP (on-line analytical processing) to user-analysts: An IT mandate. *Codd and Date*, 32:3–5.

Cuzzocrea, A., Furfaro, F., and Saccà, D. (2009). Enabling OLAP in mobile environments via intelligent data cube compression techniques. *J. Intell. Inf. Syst.*, 33(2):95–143.

Cuzzocrea, A. and Matrangolo, U. (2004). Analytical synopses for approximate query answering in OLAP environments. In *Database and Expert Systems Applications, 15th International Conference, DEXA 2004 Zaragoza, Spain, August 30-September 3, 2004, Proceedings*, pages 359–370.

Cuzzocrea, A. and Moussa, R. (2013). Multidimensional database design via schema transformation: Turning TPC-H into the TPC-H*d multidimensional benchmark. In *19th International Conference on Management of Data, COMAD*, pages 56–67.

Cuzzocrea, A., Moussa, R., and Akaichi, H. (2013a). AutoMDB: A framework for automated multidimensional database design via schema transformation. In *19th International Conference on Management of Data, COMAD*, pages 93–94.

Cuzzocrea, A., Moussa, R., and Xu, G. (2013b). Olap*: Effectively and efficiently supporting parallel OLAP over big data. In *The 3rd International Conference on Model and Data Engineering MEDI*, pages 38–49.

DeWitt, D. J., Madden, S., and Stonebraker, M. (2005). How to build a high-performance data warehouse. http://db.lcs.mit.edu/madden/high perf.pdf.

Eisenberg, A., Melton, J., Kulkarni, K. G., Michels, J., and Zemke, F. (2004). SQL: 2003 has been published. *SIGMOD Record*, 33(1):119–126.

Erik, T. (1998). Comparing different approaches to OLAP calculations as revealed in benchmarks. In *Intelligence Enterprises Database Programming & Design*.

Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pellow, F., and Pirahesh, H. (1997). Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. *Data Min. Knowl. Discov.*, 1(1):29–53.

Gyssens, M. and Lakshmanan, L. V. S. (1997). A foundation for multi-dimensional databases. In *Proceedings of 23rd International Conference on Very Large Data Bases, VLDB*, pages 106–115.

Hose, K., Klan, D., Marx, M., and Sattler, K. (2008). When is it time to rethink the aggregate configuration of your OLAP server? *PVLDB*, 1(2):1492–1495.

Hung, E., Cheung, D. W.-L., and Kao, B. (2004). Optimization in data cube system design. *Journal of Intelligent Information Systems*, 23(1):17–45.

Imhoff, C., Galemmo, N., and Geiger, J. G. (2003). *Mastering Data Warehouse Design: Relational and Dimensional Techniques*. Wiley.

Inmon, W. H. (2005). *Building the Data Warehouse*. Wiley.

Kimball, R., Reeves, L., Thornthwaite, W., Ross, M., and Thornwaite, W. (1998). *The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing and Deploying Data Warehouses*. John Wiley & Sons, Inc., 1st edition.

Kimball, R. and Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. John Wiley.

Malinowski, E. and Zimányi, E. (2008). A conceptual model for temporal data warehouses and its transformation to the ER and the object-relational models. *Journal of Data Knowledge Engineering*, 64(1):101–133.

Molina, H. G. (2013). Data warehousing overview: Issues, terminology, products. www.cs.uh.edu/∼ceick/6340/dw-olap.ppt.

Moussa, R. (2011). DDB expert: A recommender for distributed databases design. In *2011 Database and Expert Systems Applications, DEXA, International Workshops, Toulouse, France, August 29 - Sept. 2, 2011*, pages 534–538.

Nair, R., Wilson, C., and Srinivasan, B. (2007). A conceptual query-driven design framework for data warehouse. *Intl. Journal of Computer and Information Science and Engineering*, 1(1).

Niemi, T., Nummenmaa, J., and Thanisch, P. (2001). Constructing OLAP cubes based on queries. In *Proc. of the 4th ACM intl. workshop on Data warehousing and OLAP (DOLAP)*, pages 9–15.

OLAP Council. APB-1 benchmark. www.olapcouncil.org.

Papadomanolakis, S. and Ailamaki, A. (2004). Autopart: Automating schema design for large scientific databases using data partitioning. In *Proc. of the 16th Intl Conference on Scientific and Statistical Database Management (SSDBM*, pages 383–392.

Romero, O. and Abelló, A. (2009). A survey of multidimensional modeling methodologies. *Intl. Journal of Data Warehousing and Mining (IJDWM)*, 5(2):1–23.

Schnaitter, K. and Polyzotis, N. (2012). Semi-automatic index tuning: Keeping dbas in the loop. *PVLDB*, 5(5):478–489.

Surajit, C. and Umeshwar, D. (1997). An overview of data warehousing and OLAP technology. In *SIGMOD Rec.*, volume 26, pages 65–74. ACM.

Thanisch, P., Niemi, T., Niinimäki, M., and Nummenmaa, J. (2011). Using the entity-attribute-value model for OLAP cube construction. In *Proc. of 10th Intl. Conf. Perspectives in Business Informatics Research (BIR)*, pages 59–72.

Transaction Processing Council (2013a). TPC-DS benchmark. http://www.tpc.org/tpcds.

Transaction Processing Council (2013b). TPC-H benchmark. http://www.tpc.org/tpch.

Vassiliadis, P. (1998a). Modeling multidimensional databases, cubes and cube operations. In *Proceedings of the 10th International Conference on Scientific and Statistical Database Management*, SSDBM'98, pages 53–62. IEEE Computer Society.

Vassiliadis, P. (1998b). Modeling multidimensional databases, cubes and cube operations. In *10th International Conference on Scientific and Statistical Database Management, Proceedings*, pages 53–62.

Zilio, D. C., Rao, J., Lightstone, S., Lohman, G. M., Storm, A. J., Garcia-Arellano, C., and Fadden, S. (2004). DB2 design advisor: Integrated automatic physical database design. In *(e)Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 - September 3 2004*, pages 1087–1097.