Lecture Notes on "Data Warehousing by examples

Article

2 authors:

Daniel Lemire
103 PUBLICATIONS **1,719** CITATIONS
SEE PROFILE

Owen Kaser
University of New Brunswick
48 PUBLICATIONS **554** CITATIONS
SEE PROFILE

Lecture Notes on "Data Warehousing by examples"

Daniel Lemire and Owen Kaser

February 22, 2006

1 Introduction

Data Warehousing is an application-driven field. The Data Warehousing techniques can be applied to almost every type of data possible, as long as there is a lot of it.

In this lecture, we review four Data Warehousing projects and some of the insights gained by the authors. All these papers are freely available over the Web. Only one of those projects is an industrial project [1], but all are concerned with concrete problems faced by Data Warehousing experts. The topics cover include: updates from legacy data sources, qualifying the quality of a data warehouse, scalability, and performance.

We discuss the following projects:

- The WareHouse Information Project at Stanford Project (1995) [3]: the project reported on its use of monitors to check for updates in data sources.
- DWQ Project (1997): a European research project with some focus on data warehousing quality [4].
- Data Warehouse of NewsGroup (1999): an research project on the application of Data Warehousing techniques on Internet Newsgroups [2].
- Data Warehouse Population Platform (2003): recent data warehousing project by Telecom Italia with strong performance concerns [1].

2 WareHouse Information Project at Stanford (WHIPS)

The WHIPS project [3] includes monitors responsible for checking the data source for any updates (see Fig. 1). While this may appear, conceptually, as a trivial task, legacy applications often do not fire any warning when their data changes and so, the monitor might have to do a data dump and check for updates by diffing out large files. Assuming the data is in key,value format, they distinguish between 3 types of changes: update, deletion, or insertion. The point out that the problem is similar to a symmetric outer join because you are trying to match values having the same key, however, it is a bit easier since you don't have to really match the keys: you can use a deletion

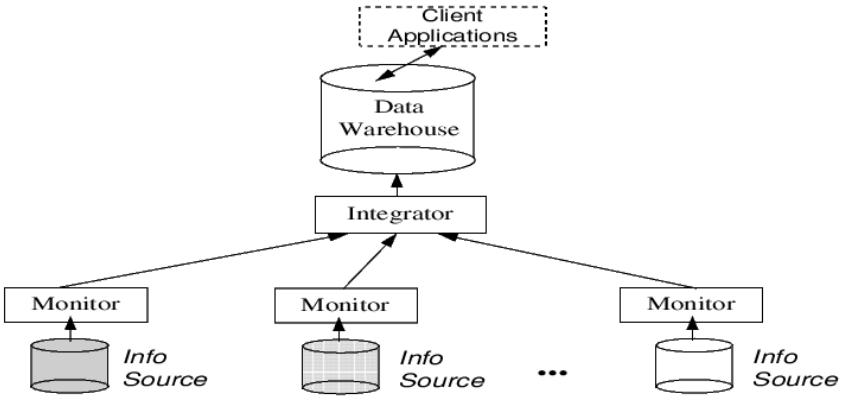

Figure 1: In the WHIPS Project, monitors are used between the Data Warehouse and the Data Source to check for changes and then forwarded to a data integrator.

and an insertion in place of an insertion. This means that instead of indexing both the new data dump and the existing data, you can stream through the new data dump and, assuming the order in which the data entries appear doesn't change too much between data dumps, you can efficiently proceed by detecting local changes in the stream of entries. This assumes you kept older data dump streams. Because you have to do the task continuously (daily), you can build buffers to help you: "The snapshot problem is a continuous problem where we have a sequence of files, each to be compared with its predecessor. Thus, in comparing F_2 to F_1, we can construct auxiliary structures for F_2 that will be useful when we later compare F_3 to F_2, and so on".

3 Data Warehouse Quality (DWQ) Project

The Data Warehouse Quality (DWQ) Project was a European project with a focus on quality in data warehousing [4]. The authors took the view that a Data Warehouse could be modelled as a buffer (see Fig. 2) and they recall that Online Transaction Processing (OLTP) are not sufficient even with high speed networks. This view of a Data Warehouse as a buffer was used in defining "quality". As a piece of trivia, they reported that in 1997 59% of respondent expected to support data warehouses in excess of 50GB, and that the average cost of projects was $3 millions.

According to the authors, measuring quality in a data warehouse can be done according to at least 4 attributes (see Fig. 3): interpretabililty, usefulness, accessibility, believability.

Interpretability do I know what the fields mean, do I know when the data I'm using was last updated?

Usefulness is the data relevant for my needs? Is the data current?

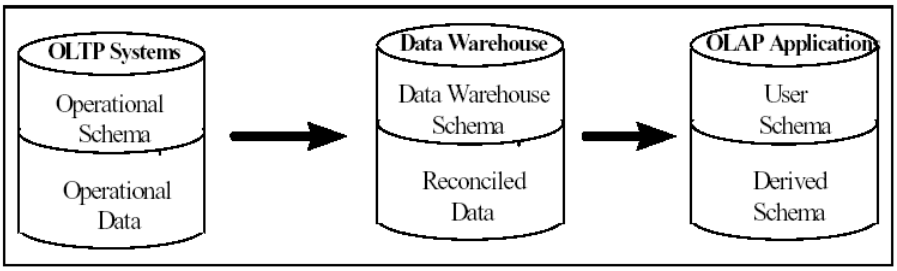

Figure 2: Data Warehousing is sometimes described as the long term buffer between OLTP and OLAP.

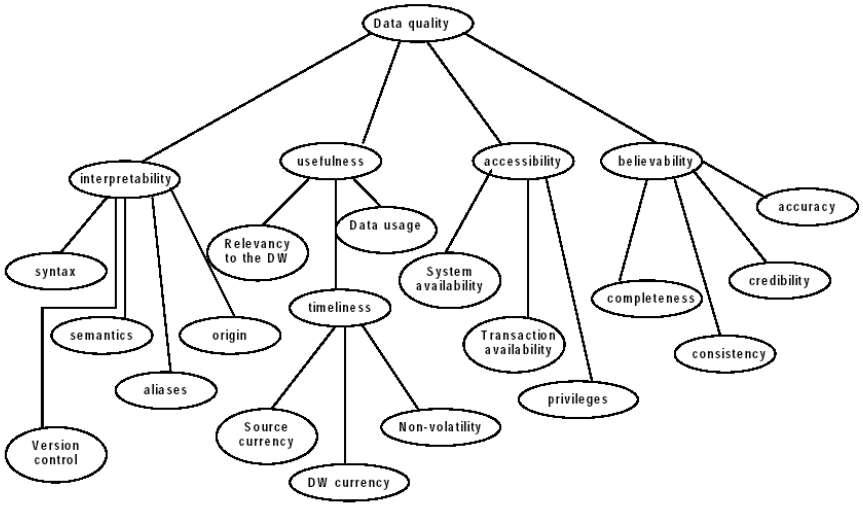

Figure 3: Some quality attributes for a Data Warehouse.

Accessibility do the people who need to have access to the data have the proper access? Is the system crashing or too slow?

Believability am I missing too much data? Are there strong biais? Is the data quality consistent?

4 The Data Warehouse of Newsgroups (DAWN)

In the DaWN [2] Project, the authors model each Newsgroup as a view over the list of **all** articles. The problem is particular because they must must support very efficiently a very large number of views. In their system, users would post post articles to the data warehouse which would then offer views to the various clients. As an example, they say that one could define soc.culture.indian as all articles in current year, similar to at

least 100 articles classified in soc.culture.indian. On the other hand, att.forsale may require user to be working at AT&T.

They say that they face two problems.

Newsgroup-selection program which views should be eager (materialized) and which should be lazy (computed on the fly)?

newsgroup-maintenance problem must efficiently insert new article into a possibly large number of newsgroup

They solve the newsgroup maintenance problem using an Independent Search Trees Algorithm using the fact that you have relatively few attributes (date, length, title, author) to your benefit: you can represent each newsgroup as a rectangular region in space and through indexes, you can quickly compute the newsgroups a given article (modelled as a point in space) belongs to. You can handle "contains" constraints using trie data structure (such as "the subject must contain the word apple".) For Body, use the frequency of words and an inverted list for fast similarility searches.

The newsgroup-selection problem should be such that frequently accessed newsgroups have faster retrieval times whereas rarely accessed newsgroup could tolerate slower access times. We can turn the newsgroup problem into a graph problem: $G = (Q \cup V, E)$ where Q are the newsgroups queries, V are the (potentially) materialized newsgroups and the hyperedges R are of the form $(q, \{v_1, \ldots, v_l\})$ where $q \in Q$ and $v_1, \ldots, v_l \in V$. Each hyperedge is labeled with a query-cost: how expensive is it to build q from $\{v_1, \ldots, v_l\}$. (Comment: For example, you can build rec.cooking.* as the union of all rec.cooking.french, rec.cooking.italian and so on, or else, you can materialized the union. On the other hand, if you have materialized rec.cooking.*, you might be able to compute rec.cooking.italian faster.) The newsgroup-selection problem amounts to finding the smallest subset M of V such that all $q \in Q$ belong to at least one hyperedge with M such that the cost is no larger than a given threshold. We can reduce the minimum set cover to a special case of the newsgroup-selection problem and so, the newsgroup-selection problem is NP-hard.

In summary, determining the views one needs to materialized in a hard problem and at least in this case, the problem is NP-hard. Sometimes the problem is not so hard (think Data Cubes).

5 Data Warehouse Population Platform (DWPP) – Telecom Italia Lab

The DWPP Project [1] was a relatively important and recent (2003) data warehousing project done at Telecom Italia. They reported that Telecom Italia collects lots of data including Call Detail Records (CDR) which are used for billing purposes and also Customer Relationship Management (CRM). Their current system can load in 20 millions CDR per hour using 2TB of storage and a machine having 12 CPUs and 16GB RAM.

One of their focus was on Extraction-Transformation-Loading (ETL) Tools. They report that none of the commerical offerings met their needs and so, they had to roll their own. They say that the most prominent task of ETL tools include

1. extraction of relevant information at the source side;
2. transformation and integration of the data coming from multiple sources into a common format;
3. cleaning of the resulting data according to business rules;
4. loading and propagation of the data to the data warehouse and/or data marts.

While it is often reported elsewhere that the common data format mentioned in the second step lies inside a relational database, their approach to ETL was to convert all data into a flat file format.

They were particularly concerned with performance and maintenance (including data integrity) at the last step in their ETL tool. It is worth mentioning that they were working with a very large data set. The main fact table, containing traffic detail data of all customers, has about 3 Billion of records. The Customer dimension table is also hash partitioned due to volumes of data (about 26 Million Customers).

All fact tables are partitioned by value and hash sub-partitioned due to:

- managing history needs - every day you have to add new partition and drop the oldest; you have to export for backup needs only the just loaded partition;
- performance needs - Parallel Query Servers work on different sub-partitions.

For increased performance, they designed a module called LDR characterized by so-called Asynchronous Read/Transform/Write with complete parallelism of read, transform and write activities (see Fig. 4). The LDR module has an internal multi-threaded structure based on in-memory structure (buffer pools) and two kinds of specialized thread groups:

Input Buffer Pool memory segments used by Readers (from Readers Thread Pool) for storing input data;

Output Buffer Pool memory segments used by ProcessUnits to write transformed data ready to be loaded by Writers (from Writers Thread Pool);

Reader Thread Pool threads that open the input data files or chunks, once that the appropriate message has been detected from the Listener, and spread that data over the Input Buffer Pool;

Writer Thread Pool threads that move transformed data from Output Buffer Pool to database tables (in direct or conventional path) or flat files.

While they did not use commercially available ETL tools, the used Oracle database, the OLAP tool was Oracle Discoverer, and the Data Mining tool was SAS Enterprise Miner.

Acknowledgement

Figures were captured from the original articles.

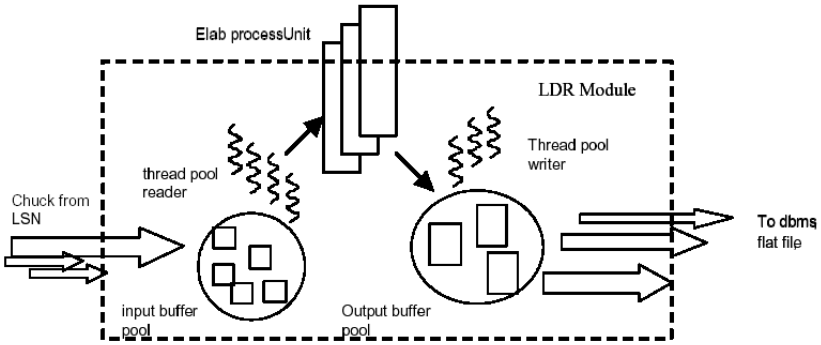

Figure 4: The loading module in DWPP (LDR) used to load and transform data into Data Warehouse tables is entirely threaded with an Input/Output Buffer Pools and Read/Write threads.

References

[1] Jovanka Adzic and Valter Fiore. Data Warehouse Population Platform. In *DMDW*, September 2003.

[2] H. Gupta and D. Srivastava. The Data Warehouse of Newsgroups. In *International Conference on Database Theory*, January 1999.

[3] J. Hammer, H. Garcia-Molina, W. Labio, J. Widom, and Y. Zhuge. The Stanford Data Warehousing Project. *IEEE Data Engineering Bulletin*, 18(2):41–48, June 1995.

[4] M. Jarke and Y. Vassiliou. Data Warehouse Quality Design: A Review of the DWQ Project. In *Information Quality*, 1997.