Advances in DataWarehouse: A Survey

Tonáhtiu Arturo Ramírez Romero1, Gumersindo David Fariña López2, Claudia Lizbeth Martínez González3

1Instituto Politécnico Nacional, México, tonahhtiu@yahoo.com
2Centro de Estudios Científicos y Tecnológicos No.7, Instituto Politécnico Nacional, México, dafarina@hotmail.com
3Sección de Estudios de Posgrado e Investigación, Escuela Superior de Ingeniería Mecánica y Eléctrica, Instituto Politécnico Nacional, México, C. P. 07738, lizbeth.martinez@gmail.com

Abstract

In this work we present a brief introduction to data warehouse, followed by the most relevant topics in the last years. First shown Knowledge warehouse, which is a data warehouse for data sources formatted or unformatted, later we present Data warehouse Federation, It focuses to interconnect different systems established by access rules, after we present Near Real Time – ETL which aims to update the stores quickly and close to online systems, later DW quality and finally QuiDA Architecture

Keywords: Data Warehouse, Near Real Time, Knowledge warehouse, Federations

Introduction

Data warehouses have been used for a long time; after several advances, in the actual complex scenario, this survey is directed to those who want to approach this discipline and contribute to its development. Begins by giving a brief introduction about data warehouses, followed by the most relevant topics in the last years. For convenience, data warehouse is hereinafter abbreviated as DW.

A concrete definition of a data warehouse is given by Inmon (2005): “A data warehouse provides an infrastructure that enables businesses to extract, cleanse, and store vast amounts of corporate data from operational systems for efficient and accurate response to user queries”, without placing a burden on the operational source systems.

Data warehouses are widely recognized as decision support systems for the analysis of huge volumes of alphanumeric modeled data using a multidimensional model, which defines the concepts of facts and dimensions (Kimball, 1996). The facts of a data warehouse are the values of the indicators to be analyzed (Malinowski & Zimányi, 2008).

The basic purpose of a DW is to empower the knowledge workers with information that allows them to make decisions based on a solid foundation of facts (Devlin, 1996). DW mechanism is the following; data is extracted from the sources, subsequently transformed and cleansed and...
finally loaded. The data warehouse and data marts are SQL based databases systems (Amar & Dan, 2011). The data mart is the layer used to access the data warehouse.

Evolution in business processes influences DW evolution, which has been studied from different approaches; data source schema, DW schema, and analytical needs (Taktak, Feki, & Zurfluh, 2014).

DW technology has evolved in many trends, during this evolution important aspects have emerged: Knowledge Warehouse, DW as databases using Hadoop, DW federations, Near Real Time ETL, DW Quality, QuiDA Architecture, among others. In this survey we show the most important of these trends.

Knowledge warehouses
Nemati et al. (2002) proposed the Knowledge Warehouse architecture, hereinafter called KW. KW is a variant of DW, but focused on data from various sources like strings, production rules, binary, objects, mathematical models, and what-if cases. A KW load, transform and catalogs different types of knowledge (i.e. both tacit and explicit).

The enterprise data warehouses can be extended to create knowledge warehouses (KW). KW is based in the knowledge spiral: socialization, articulation, integration and understanding/internalization.

Socialization is the process of teamwork sharing with others the experiences, technical skills, mental models, and other forms of tacit knowledge. Articulation is the conversion process of tacit knowledge to explicit knowledge. Integration is the process of combining, explicit representations of knowledge for new relationship and patterns. Understanding is the process of evaluating and validating the new relationship and patterns in the context.

Considerations for knowledge warehousing.
The user is an expert in decision making, not in technology. To extract information from unstructured sources, data mining is an option, as well as other Artificial Intelligence techniques. The expert user is able to use an array of analytical tools to work with the knowledge spiral.

The KW must have the following capabilities.
• Retrieve, store and use in diverse forms
• Manage the analysis task, with limited interaction regarding cognitive aspects from the decision maker.
• Manage feedback loop, to improve results.

Nemati et al. (2002) proposed the architecture for KW which basically consists of four modules: Data and Knowledge extraction/acquisition; Data and Knowledge Transformation & loading; Knowledge Warehouse (storage); Knowledge Analysis Workbench and the correspondent interfaces.
This proposal architecture is big and not well delimited, authors explain that development and implementations was not possible because this may involve considerable amount of organizational time and effort and could cross the boundaries of many business units and departments.

By other hand, ontology has also been proposed to implement algorithms to integrate DW schemas and query results in the case of semantic heterogeneity (Naoual & Abderrafliaa, 2014).

**Dataware House Federation.**

A federation of data warehouse is a set of data warehouses. The federation is usually distributed in different physical locations. The main reason of forming federations is to reduce costs of such an operation; instead of making one centralized system, it is easier and faster to create a loosely coupled network of existing data warehouses (Jindal & Acharya, 2004; Waddington, 2004).

The management system for a data warehouse federation consists of an user interface enabling presentation of user queries, a program for query decomposition and a program for integrating knowledge coming from different data warehouses as the answer to a user query (Kern, Ryk, & Nguyen, 2011).

A federation management system contains the following elements (Kern et al., 2011):

1. An integration procedure of the schemas of the component warehouses giving the logical schema of the federation.
2. A query language for the user who does not need to know the schemas of the component warehouses.
3. A procedure, which enables decomposition of user queries to the federation into sub-queries, which are sent to the component warehouses.
4. A procedure integrating the answers sent from the component warehouses to federation layer.

A federation has a layered structure with two layers: component and federation. The federation layer contains the federation schema, is a simple interface to be used by other applications hiding heterogeneity of components (Berger & Schrefl, 2008; Schneider, 2006). Component layer contains information for each local DW.

A data warehouse federation works only on logical level query processing; processes are performed in its components (Sharma, Goswami, & Gupta, 2009).

In Figure 1, a DW federation diagram is shown, ETL means Extract Transform Load Tools.
Other possibility in the field of DW, integrate and interface relational database and data warehouse systems. This is possible using different tools and technics (e.g. Hadoop (Apache, 2015; Cohen, Dolan, Dunlap, Hellerstein, & Welton, 2009), MapReduce (Apache, 2016a), HIVE (Apache, 2016b)), these tools are part of the Apache Hadoop project (Das & Mohapatro, 2014).

Hadoop is a free computational tool of MAD kind (Magnetism - Agility - Depth) (Cohen et al., 2009). Java-based programming framework to interact with large storage systems or big data, Since it is focused on attracting different data sources, it has the agility to adapt their engines to changes in data sources and can act in depth in these sources, and much more beyond the possibilities of traditional SQL-based analysis tools (Cuzzocrea, Song, & Davis, 2011). Hadoop system relies on Hadoop Distributed File system (HDFS) that provides high-throughput access to application data. Hadoop can use the data warehouse as a data source. An open source data warehousing solution built on top of Hadoop is Apache HIVE, which facilitates querying and managing large datasets residing in distributed storage. Hive provides a mechanism to project structure onto this data and query the data using an SQL-like language called HiveQL (Apache, 2016b).

MapReduce is a program that can generate and process large data sets, this tool works on the top of the HDFS which was developed as an abstraction of the map and reduce primitives present in many functional languages (Apache, 2016a; Dean & Ghemawat, 2008). We can say that a MapReduce is just another program and the DW is just another database, in this context MapReduce can generate SQL statements to the DW.

Cheetah is other DW specifically designed for advertising online, it works on the top of MapReduce to allow various simplifications and custom optimizations (Chen, 2010).
Near Real Time – ETL

As source data changes over time, periodical updates over DW are necessary; this process usually takes place in the hours of less load on the operational system. The full update process is referred as cycle. The action of shortening the data warehouse loading cycles is referred as Near Real Time data warehousing (RTDW) or microbatch ETL (Kimball & Caserta, 2004). Processing is achieved in RTDW.

DW usually have auxiliary tools that allow them to interact with the operational system, they are called Extract Transform Load (ETL) tools or ETL system. The first population of a DW is referred to as initial load. During this process, data from the sources is fully extracted, transformed, and load in the DW; these tools can also update the DW.

There are two options to update the DW, incremental loading and full reloading. Full reload is the common process but is inefficient by the traffic generated on the network; on the other hand, propagating just the changes to the DW is known as incremental loading. This approach is preferred because it generally reduces the amount of data to process.

An alternative to update a data warehouse is temporal data warehousing, where changes are stored by inserting new rows with the changes and expiring the old versions, while deleted rows are labeled as expired and not deleted (Rahman, 2014).

For the particular case of the operational sources, there are mechanisms that support the incremental loading to interact and receive information of the changes that have occurred in it, this mechanism is called Change Data Capture (CDC). CDC has many methods for example, snapshots sources, logged sources, database log scraping, log sniffing, time stamped sources and lockable sources (Jörg & Dessloch, 2014).

Figure 2. Interface between Hadoop and Data Warehouse (Das & Mohapatro, 2014)
Snapshot sources. This method provides the system status information at the time of data extraction. In this method the ETL system uses an intermediary working area called staging area, to recover information about the operational system, the inconvenient is the large amount of space to store it.

Logged sources. This method is related to the sources of data used files or tables called logs, where a system state is automatically or manually stored. This is done among other methods through triggers, which are called automatically synchronous logs and asynchronous logs. From the use of these logs, two techniques arise: Database log scraping and log Sniffing. Scraping refers to the analysis of log files; sniffing refers to the analysis of the information at the time it is reached.

Timestamped sources. In this method some systems store the date and time of change or insert; the problem with this method is that it is not possible to know which log was cleared.

Lockable sources. The database manager commonly uses this option; it refers to block the operations that generate changes in the data sources.

However, it has been observed that during the update process, a possibility of warehouse refreshment anomalies exists (Zhuge, García-Molina, Hammer, & Widom, 1995). The refreshment anomalies are caused by two reasons, the ETL extract data while the source is changing and the ETL do not consider a change in the structure of the database tables. There are several approaches to prevent data anomalies.

Zhuge et al. (1995) proposed the Eager Compensating Algorithm (ECA) for a single remote data source and a Strobe family of algorithms designed for multisource environment. The basic idea behind both is to keep track of source changes that occur during DW refreshment and perform compensation to avoid the occurrence of anomalies. Both algorithms assume that the software will notify about the changes as soon as they occur.

Jörg & Dessloch (2014) proposed a solution that neither lock operational sources nor maintain data copies in the staging area. First, the ETL system can prevent to detect the lack of consistency in the data modified and second, uses advance change propagation (Jörg & Dessloch, 2014).

DW quality
A significant issue to evaluate a DW is quality, divided in data model quality, presentation quality and data quality. DW conceptual data models are considered one of the most relevant features to assess general quality; subsequently, different approaches have been proposed as metrics to estimate them. Some proposed metrics are star, schema, table, class scope, diagram scope and dimension hierarchy metrics, among others. Although quality factors have not been standardized, characteristics such as understandability, complexity, maintainability and effectiveness have been used as the main proposed factors, while the first one, understandability is the most common quality factor. To validate these factors, empirical and theoretical validation
must be achieved and different techniques have been proposed. The most used techniques for empirical validation are regression analysis (for parametric test techniques, Pearson correlation is the most used; for non-parametric tests, Spearman’s Ro the most used) and diverse machine learning techniques (fuzzy logic, decision trees, Bayesian classifiers). By other hand, for theoretical validation, axiomatic is the most used technique. The utility of some metrics such as legibility, effectiveness or correctness have not been analyzed (Heena, 2015).

**QuiDA Architecture.**

This proposal focuses on quality requirements within an environmental information system. The proposed architecture can be used to transfer data quality indicators from input databases to the spatial data warehouse, integrating quality dimensions within the warehouse. One of the proposed metrics is the use of syntactic ratio correction. Data quality is integrated into the database, including a data quality model, quality and data are extracted and loaded in the cube, the DW can be used with data mining methods (Berrahou et al., 2015).

Evolution in the field of DW continues, here we have presented the topics that were considered most relevant.

**Acknowledgements**

We like to express sincere appreciation and deep gratitude to all participants in this work. Authors kindly acknowledge financial and computational infrastructure support from Instituto Politécnico Nacional to facilitate the development of this survey.

**References**


