CONSIDERATIONS ON REAL TIME DATA WAREHOUSING (RTDW)

Marius Bogdan DINU, PhD Student
University “POLITEHNICA” of Bucharest, Faculty of Electronics, Telecommunications and Information Technology, Department of Electronic Devices, Circuits and Apparatus

Abstract: The RTDW concept originated in the early 2000s. By that time, computing power had increased to a level that was allowing extraction of data collections for reporting purposes. Such collections were used almost in real time and at speeds nearly comparable to what an operation system was capable to deliver. The main idea will be to eliminate some of the components of the classic extraction process which is basically the most costly factor less time-consuming. We anticipate that the following factors will be decisive: elimination of batch-type processes [1], data compression techniques, data capture techniques, ability to keep in cache a large volume of data, parallel processing, and data mining algorithms that can adapt to such applications.

Keywords: data compression technique, data capture technique, large volume of data, data mining, data warehousing, snapshot

1. INTRODUCTION

In approaching the RTDW (Real Time Data Warehousing) [2] concept, the following two definitions may be considered: firstly, RTDW may be defined as a form of aggregation of analytical data in a data warehousing environment using a continuous process or in an almost time real loading. Secondly, RTDW may be defined as data storing and analyzing in a continuous manner and in a particular storage environment.

Based on the definitions mentioned above, any new event relative to a change in data (such as update, insert or delete operations performed on various data sources) should become available for reporting in the fastest way possible or, if possible, almost instantly. One restriction is that original source-systems should remain computationally unaffected or as little requested as possible. In other words, though data mining is continuous, the source-systems should have the same operational capacity at all times.

While the classical DW (Data Warehousing) is designed to collect historised (instant-snapshot) information and to later on generate reports meant to support the management’s decision-making, this new concept is conceived to obtain an immediate response[3] to the most recent information that is continuously supplied by the trading systems[4]. Below we present a comparison between the two approaches.

2 CLASSIC DW VERSUS RTDW

According to WH Inmon [5], a data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process. By contrast, RTDW is based on collection and use of the latest information. More specifically, while data change (are updated) constantly, the purpose of this system is not to store historical data, as it happens with DW, but to concentrate data collection exclusively on the most recent state of the transactional systems.[6]
Highlighting further the differences between the new system and the classic DW, we may say that the latter is directly influenced, in terms of working times; by the ETL (Extract Transform and Load) processes. Seen from this perspective, the ETL cycle is a generic description of the following workflow: data are first of all extracted from external sources, brought to a landing area (staging area) and are then compared to a specific model. Subsequently, data are uploaded in specific tables (star schema); these writings are done based on clear transformation rules and in combination with specific restrictions applicable to the industry segment. After their consecutive processing, data will finally reach the reporting area.

In another train of thoughts, with the classical DW system, an ETL-type of process uses effective mechanisms to process data, such as bulk insert, parallel updating or other methods of effective handling in parallel of computational resources. Although such techniques are highly advanced and fully automated, they take long execution times. This is typical for batch-type processing operations (that are common with most of the classical DWs). Unlike OLTP (OnLine Transactional Processing) systems, batch processing operations are not interactive\(^1\). In other words, the completion of an operation within the ETL will not automatically lead to availability of data for reporting.

3. CLASSICAL SCENARIOS AND SOLUTIONS IN RTDW

As we have explained before, for transferring data over the network solutions should be found to accelerate such operations in the RTDW. Assuming that a certain amount of data extracted from sources from 1 to N is too small versus the reporting needs, then the use of such type of iteration would be out of question. But what happens when a large number of Gigabytes is requested over the network for a continuous compilation of reports?

To simplify the scenario, we assume that the entire system is properly configured, has enough CPUs, is properly balanced in terms of inputs and outputs and allows the running of processes in parallel. Basically, what is important in this case is to determine the correct strategy for data transfer through the network.

Suppose we have a scenario where two tables, A and B, must be transferred over the network, each of which exceeds 10 Gbytes and 10-20 million lines. Starting from the premise that we cannot enlarge the transfer bandwidth, we can nevertheless solve this problem by means of compression techniques. The main idea is that only a limited number of bits will be copied and transferred through the network. Of course, this artifice will require specific computational power, in order to be able to exploit real-time data. At a compression ratio of 7.7 x, for a total of 1057 Gb we will have a transfer demand of about 136 G, so the 20 Gbytes could be transferred and used at an acceptable speed of a few seconds (or even faster, where better compression techniques and sufficient hardware resources are available).

Normally, in a classic DW data are loaded in a staging area. There, the data are compared with predefined values related to the type of business and the industry segment, but they are also conditioned by the level of granularity; later on, the date undergo changes in order to be adapted them to the star-type schemes where the lines are finally stored. With a RTDW, such intermediate data layer is inexistent.

\(^1\) http://www.damanconsulting.com/company/articles/dwrealtime.htm
All lines are recorded in the final fact tables with only a minimal processing effort.

From another perspective relative to RTDW, there will always be a mechanism for capturing the information between the transactional area and the real time reporting area. For the capture mechanism to operate properly, both the reporting information and the reporting environment must be perfectly synchronized. At the same time, the transactional systems from where data are collected should operate at a constant level of performance.

There are several methods to solve issues dealing with synchronization and data delivery in real time, namely:

a) **CDC Log** (*Change Data Capture using Log*). This mechanism is considered to be the most powerful. On one hand, data sources are not affected when data extraction occurs. On the other hand, data delivery speed is very good. The principle used is linked to the fact that the vast majority of the data servers incorporate data backup and recovery mechanisms, as well as journaling systems. One method is that where all changes are recorded in log files. Basically these contain the entire history of the changes using the commit strategy (specific to each technology).

b) **DBMS triggers**. These objects can include code segments that are associated with a table, a scheme, a database or any other program or subprogram. When certain conditions are met, these triggers are executed. Special conditions may include time related aspects (usually in RTDW), but, at the same time, complex interrogations can be added in order to retrieve only a subset of data that is relevant for reporting. In terms of information flow, this extraction mode works as follows: when all the conditions are fulfilled, sets of lines are inserted or updated in the tables dedicated to reporting.

c) **Time samples**. Many operating systems keep data in specific columns indicating the time when the last change occurred in relation to certain lines. Because the system allows the checking of these fields, extraction may be conducted at well-defined points in time. This particularity of some systems can be used for reporting. However, there are disadvantages when some transactions lead to deletion of some of the lines. In this case, some of the extraction criteria may be lost and the reporting may be erroneous.
d) Message tails. Another method is to capture the changes directly from the application, and not from the database (IBM MQ, for example, uses this method). Changes are reported and produced by tools of the ETL type. Yet, there are two potential problems: on the one hand, an additional cost will be incurred (license cost), and, on the other hand, the transactional systems will experience cases of manual reconciliation.

e) Mirrored transactions. We start from the assumption that we know what type of transactions is active and we have a mechanism that allows us to write the content (of the active transaction) to an external table. Unfortunately this mechanism cannot be used incrementally nor can it be used for determining a full database context (as it is a capture based solely on transactions).

Whatever the capturing mechanism we use, the results will be written towards the fact tables. After a while this information will go through an “aging” process whereby a line (or a set of lines ), which is the subject to real time reporting, will, after some time, become informationally obsolete and is no longer used. As such, it is important for practitioners to try to keep in cash in the memory (only) the active data set. When the set in question becomes static (receives no more current interrogations), it can be written on the disk. Meanwhile, another set becomes active and enters the same aging cycle.

With or without capture mechanisms, when we have to deliver reports in real time, the degree of parallelism may be another key-factor. Regardless of the type of insert/update type of cycle we may use, if we want to obtain good results, we will have to either build our own parallel operation mechanism or use the options made available by existing platforms.

When a parallel extraction mechanism is employed, we should take into account several restrictions that are governing the process. If, for example, we plan to use for extraction purposes a SQL having imbricated hints of parallel operation type, and assuming that there is a set of parallelism-related parameters already correctly set and a number of sufficiently powerful CPUs, we might encounter problems, if data are not distributed on a single hard disk. In this case, we will see that the parallelism hint will not work because, basically, all the criteria on the disk are serial and will be executed by the disk controller, which is a singular device. A correct method is to distribute the storing over several disks.

4. THE QUALITY AND THE PRE-PROCESSING OF DATA IN RTDW

As we have shown above, data delivery speed is an extremely important factor in RTDW. One possible question is whether some aspects concerning data quality might be neglected for the sake of a decrease of the processing times. The right answer to this question seems to be a contradiction in terms. Yet, in the real world we cannot avoid restrictions dealing with data quality. That is why an important amount of time will be dedicated to tackling this topic, so that we may assure data quality and be able to perform cleansing /pre-processing operations (where the context so requires).
The processes designed to ensure data quality must address important issues such as data consistency, accuracy, and avoidance of duplication of results. Results obtained in the absence of proper quality of the data might be inadequate or generate counterproductive situations such as, for example, further reconciliations, increased costs, possible data transfer errors, incorrect data remaining in the model, less confidence in the reporting system. Likewise, the data preparation process is a concept covering all the actions required to satisfy entries in any complex data processing algorithms.

Any type of algorithm proposed to be used for future processing operations must have all these pre-processing already solved; otherwise the following critical situations may occur: inadequate values transferred to algorithms, wrong results of algorithm running or, simply, algorithm failure. To conclude, the proposal to increase the RTDW speed in the detriment of data quality check cannot be accepted as a viable working hypothesis.

5. POSSIBLE RTDW SOLUTIONS USING DM ALGORITHMS

We have tried to present some of the techniques already considered as classic approaches in terms of the RTDW perspective. A key question is whether we can adopt technologies that are more productive. A potential answer may be the utilization of data mining algorithms.

Besides the high performance obtained by means of these algorithms, complementary information can also be obtained, such as treatment/detection of incorrect data, quick localization of errors, a procedure to replace data that do not match with the model, distinct reports containing atypical records (if any).

A classic definition of the data mining process is a practice that automatically searches in the large databases area to discover new forms. For this purpose, algorithms are used, that are able to perform data segmentation or to determine subsequent events. An important mention is that data mining processes can solve problems which cannot usually be resolved by the simple running of an interrogation or by means of normal reporting techniques. Some of these algorithms include Decision Trees, Neural Network, Naïve Bayes, K-Means, Non Negative Matrix Factorization, Support Vector Machines, O-Cluster etc. We know that data mining algorithms require a high degree of computational power usage and therefore they can be expensive when we already have a number of time restrictions.

For example, in situations such as the keeping a large lot of data in memory or in cases where data preparation requires the doubling of the preprocessing time, we may see that, in such combinations, any data mining algorithm cannot be run from the RTDW perspective. Yet this is not always the case. There are situations where such problems can be solved by specific data mining strategies. Thus we can select only a subset of data, so as to avoid the processing of the entire set (with the reuse of partial data); we may run the algorithm distributed on several machines; or, if we have enough computational power, the entire algorithm can be run on a data set that is complete in terms of RTDW requirements. Using data mining algorithms for real time reporting remains still a difficult adaptation process.

Each data mining algorithm is fully described by constraints, limitations and strengths. All these algorithms are dedicated to solving three major categories of problems: classification, 

clustering and prediction. The fact is that the difference between clustering and classification is
given by the dissociation criterion. Classification implies the existence a clear set of separation
rules, whereas with clustering there are no distinguishing criteria.

With regard to classification, the key factor is the identification and selection of a
particular algorithm, which is able to grow faster with every iteration. This means that after a
number of drive iterations, the algorithm will become fast enough to be a component of the
RTDW, without any influence on the performance.

A possible example would be SVM (Support Vector Machines)\(^3\) classification. This
algorithm is defined as a method used in classification and regression. Using a first set of data
for model drive, we create a model which, in the case of binary classification, will work at a very
good speed. Any new entry generated will be instantly classified into two subsets.

Mathematical considerations aside, we may emphasize the fact that this algorithm has a
good accuracy, a good classification speed and a good tolerance to missing values. As a
downside, we should mention the low “learning” speed, the difficult treatment of parameters, the
need to preprocess data (which can be a problem from an RTDM perspective), intensive use of
CPU and memory. In the case of data that are non-linearly separable (the separate line is the
Gaussian line), the volume of data generated is very high and, hence, classification cannot run at
very high speeds.

6. CONCLUSIONS

To summarize, we may say that it is difficult to determine which would be the decisive
factor in the future development of RTDW. Still, there are a few factors that may be considered
as having the potential to influence the concept:

a) Data compression technologies can reduce network transfer times and may be
combined with capture mechanisms based on logging.

b) A new RTDM paradigm will change the technical approaches to advanced analytical
methods and it may generate better data protection, extract new inexistent
information, generate predictions to improve delivery times.

c) The capacity to process in parallel large volumes of data. As we have explained
above, we are no longer dealing with the classic ETL process; yet the minimal
processing form that remains (at least for data quality assurance) should be able to
execute in a parallel a large number of operations.

d) Prediction\(^4\)- combining ETL mechanisms with prediction may result into reduced
calculation, and replacement of missing values. Similarly, for new attributes, new
values can be determined.[7]

e) Data partitioning is costly from the writing standpoint and therefore it is a weakness
of the RTDW. But, in terms of reporting, such an option will increase performance
and allow execution of parallel interrogation.

---

\(^4\) http://blog.bigml.com/2013/03/12/machine-learning-from-streaming-data-two-problems-two-solutions-two-
concerns-and-two-lessons/ Charles Parker
f) RTDW does not conceptually fit into the classical model of snapshot collection from DW.

g) Batch-type processing cannot be used with RTDW, but there will be new forms of ETL for fast data processing.\[8\]

All in all, the subject is difficult to be fully estimated or evaluated in terms of its future development. Chances are that we will witness the emergence of new technologies that will serve as an important support for the RTDW.

7. REFERENCES