An Approach for Alert Raising in Real-Time Data Warehouses

« Journées francophones sur les Entrepôts de Données et l'Analyse en ligne »
Brussels, Belgium, April 2-3, 2015

Maximiliano Ariel López*, Sergi Nadal Francesch**
Mahfoud Djedaini***, Patrick Marcel***, Verónika Peralta***, Pedro Furtado ****

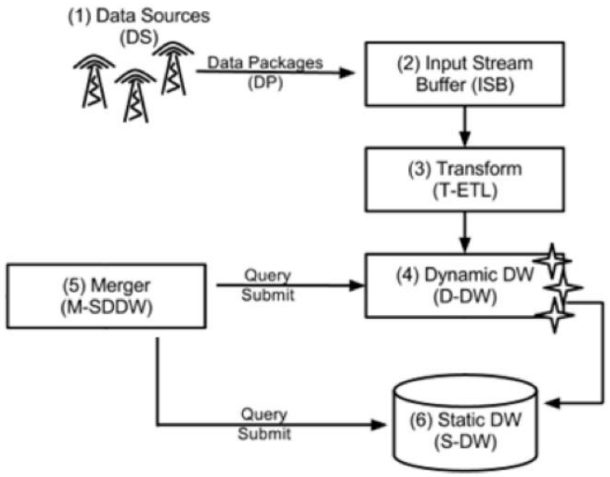
* École Centrale Paris
** Universitat Politècnica de Catalunya
*** Université de Tours
**** University of Coimbra

Introduction

- Currently, many organisations have the requirement of analysing their information in a real-time manner:
  - Energy Production and Consumption
  - Traffic Monitoring
  - IT Networks Monitoring
  - Stock Markets
- Monitoring and quickly detecting deviations from the expected behaviour allow analysts to face abrupt changes.
- To enable near real-time analysis based on the most recent information, data warehouse architectures have been extended or adapted.

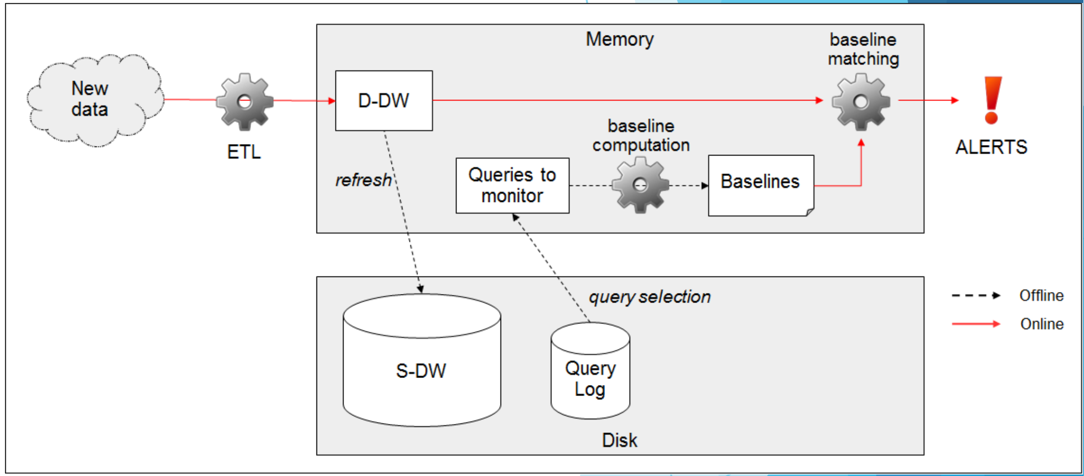
Real-Time Data Warehousing

- Ferreira and Furtado have proposed an approach that implements a real-time data warehouse without data duplication which is composed of three main components:
  - the Dynamic Data Warehouse (D-DW),
  - the Static Data Warehouse (S-DW) and
  - the Merger.
- In our paper, we present an approach for alert raising in a real-time data warehouse that assumes this architecture.
- The key idea involves leveraging query logs to build an in-memory summary of the S-DW and then checking this summary against the data in the D-DW to raise alerts.
- We assume that user traces express sets of facts that need to be monitored.

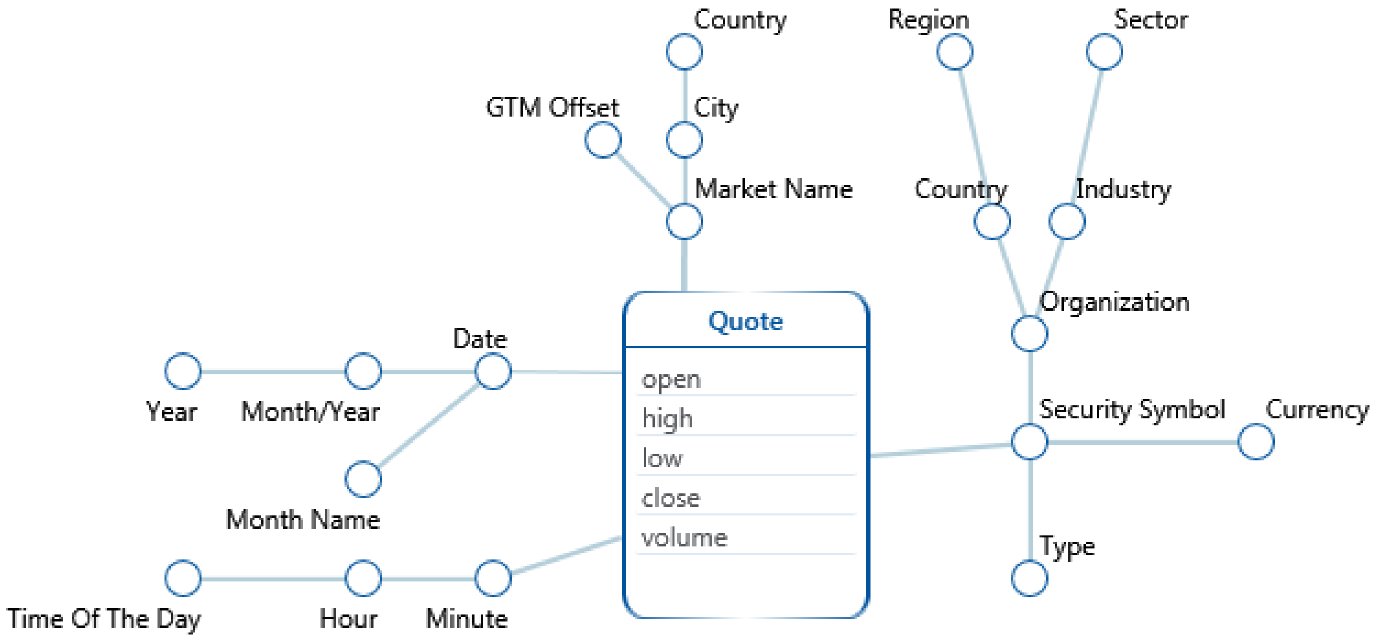


Proposed Approach

- In an offline phase, for each query, we construct a "baseline":
  - The query is run over the S-DW.
  - A confidence interval is calculated for the facts contributing to each cell.
- Confidence intervals are built using the bootstrap method (Efron and Tibshirani, 1986).
- This method is particularly well adapted to a real-time context:
  - Unknown population: complete answer of the query.
  - Sample: current answer to this query.
- In the online phase of our approach, new data loaded into the D-DW are compared to the appropriate baselines. This comparison is used to raise alerts.



Stock Exchange Markets Example



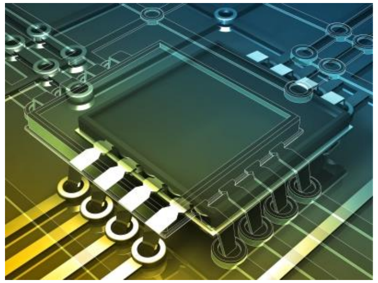
Period	Data Volume per Security
Up to 50 years ago	1 record every quarter
Up to 20 years ago	1 record every month
Up to 10 years ago	1 record every week
Up to 3 years ago	1 record every day
Up to 15 days ago	Around 100 records every day
Up to 1 day ago	Around 400 records

- New York Stock Exchange (NYSE)
- National Association of Securities Dealers Automated Quotations (NASDAQ)
- Buenos Aires Stock Exchange (MERVAL)
- Mexican Stock Exchange (IPC)
- Sao Paolo Stock Exchange (BOVESPA)
- Currency Exchange Rates

Example: Starting Point

Log Example:

	Group By Set	Filters	Measures
Q_1	[Market.Geography].[Market Name] [Security.Type].[Security Symbol]	[Security.Geography].[Organisation].[Google Inc.]	[Close]
Q_2	[Security.Geography].[Organisation] [Market.Geography].[Market Name]	[Security.Activity].[Sector].[Health Care] [Security.Geography].[Country].[USA]	[Open], [Close]
Q_3	[Security.Activity].[Security Symbol] [Date.DateMonthYear].[Year]	[Market.Geography].[Market Name].[NASDAQ] [Sector.Activity].[Industry].[Semiconductors]	[Volume]
Q_4	[Security.Activity].[Security Symbol] [Date.DateMonthYear].[Year]	[Sector.Activity].[Industry].[Water Supply]	[All]



Example: Baseline Computation

History Cube - Query Results

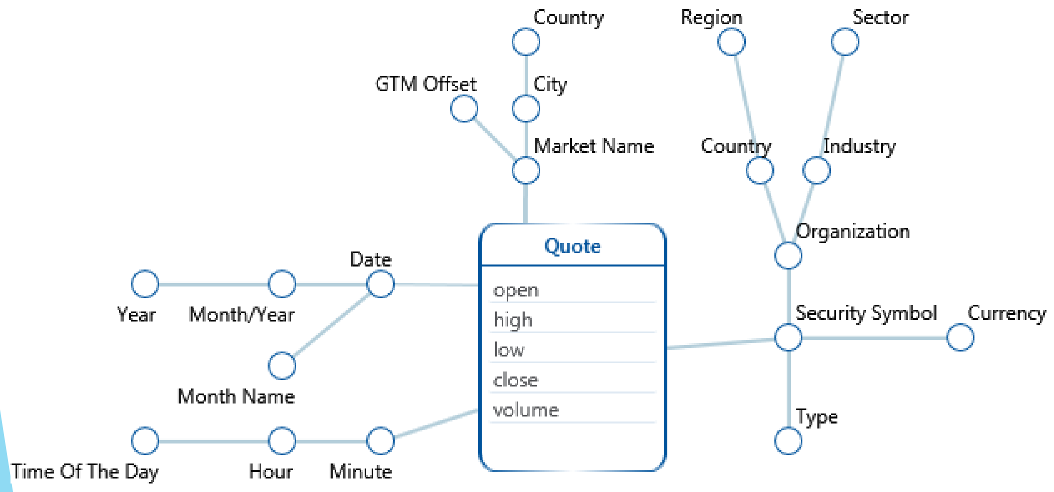
Market Name	Security Symbol	Close
NASDAQ	GOOG	569.721
NASDAQ	GOOGL	441.326

date_id	time_id	market_name	security_symbol	open	high	low	close	volume
20141009	2259	NASDAQ	GOOG	559.0600	571.4...	571.1...	560.8800	517900....
20141009	2000	NASDAQ	GOOG	560.8800	560.8...	560.8...	560.8800	.0000
20141009	1959	NASDAQ	GOOG	561.0000	562.3...	561.5...	561.0500	17000.0...
20141009	1954	NASDAQ	GOOG	560.1200	561.6...	560.5...	561.5100	3900.0000
20141009	1949	NASDAQ	GOOG	560.6800	562.0...	561.8...	560.9200	6700.0000

...

date_id	time_id	market_name	security_symbol	open	high	low	close	volume
20141009	2259	NASDAQ	GOOGL	569.0300	582.5...	581.6...	570.8100	411700....
20141009	2000	NASDAQ	GOOGL	570.8100	570.8...	570.8...	570.8100	.0000
20141009	1959	NASDAQ	GOOGL	570.7300	572.2...	571.4...	571.1100	66300.0...
20141009	1954	NASDAQ	GOOGL	570.4200	571.6...	570.6...	571.3000	8000.0000
20141009	1949	NASDAQ	GOOGL	571.0000	572.0...	572.0...	571.0000	3600.0000

...



- Boostrap replications (e.g. 100 or 1000)
- Sample percentage (1 %)
- 95% confidence rate:
  - Percentile 2.5
  - Percentile 97.5

Example: Persisted Baselines

Baselines Viewer

Baseline (Header)

Cube	Group by set	Filters	Cells #	Id
Quote	[Market.Geography].[... [Security.Type].[Securi...	[Security.Geography].[...	10	61

Cells (Items)

Coordinates	Measure	Min. Mean	Max. Mean	Min. Dev.	Max. Dev.
[Market.Geography].[Market Name].[NASDAQ] [Security.Type].[Security Symbol].[GOOG]	[Measures].close	568.586	571.243	8.498	12.878
[Market.Geography].[Market Name].[NASDAQ] [Security.Type].[Security Symbol].[GOOGL]	[Measures].close	423.475	464.039	144.287	163.634
[Market.Geography].[Market Name].[NASDAQ] [Security.Type].[Security Symbol].[GOOG]	[Measures].high	570.212	572.516	8.054	11.987
[Market.Geography].[Market Name].[NASDAQ] [Security.Type].[Security Symbol].[GOOGL]	[Measures].high	427.706	468.189	143.079	161.332

Interval for GOOG with 2 standard deviations:

- Lower bound: (568.586 – 2 * 12.878) = 542.83
- Upper bound: (568.586 + 2 * 12.878) = 594.342

Interval for GOOGL with 2 standard deviations:

- Lower bound: (423.475 – 2 * 163.634) = 96.207
- Upper bound: (423.475 + 2 * 163.634) = 750.743

Motivating Example (cont.)

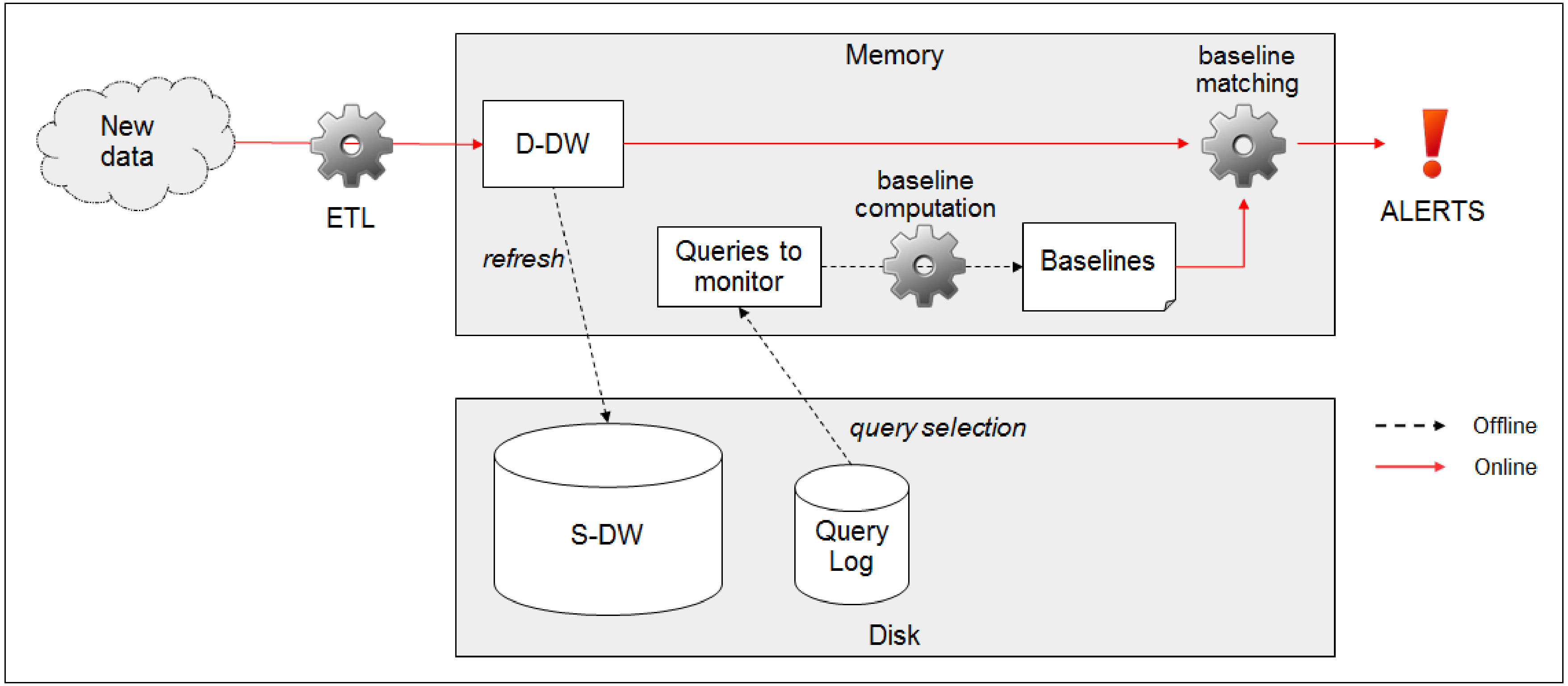
Baseline Example for Close measure (Q_1):

NASDAQ	GOOG	[542.83 – 594.342]
NASDAQ	GOOGL	[96.207 – 750.743]

The following fact inserted into DDW might then trigger an alert:

date_id	minute_id	market_name	security_symbol	open	high	low	close	volume
…	…	NASDAQ	GOOG	…	…	…	542	…

Example Recap



Baselines Refresh

$$\frac{|DDW_Q|}{|DDW_Q|+|SDW_Q|} \times (1-(1-s)^b) \times (1-(1-\frac{1}{|Q|})^{|DDW_Q|+|SDW_Q|})$$

- "Q" is the query from which baselines are derived.
- "DDW_Q" is the set of facts of the real time component of the DW covered by Q.
- "SDW_Q" is the set of facts of the history component of the DW covered by Q.
- "s" is the sampling percentage
- "b" is the number of bootstrap replications.
- $\frac{|DDW_Q|}{|DDW_Q|+|SDW_Q|}$ is the probability that a fact comes form the real time component.
- $(1-(1-s)^b)$ is the probability that a fact is chosen for the bootstrap computation.
- The last term is the probability that a cell of the baseline covers at least a given primary fact, which is derived from the Cardenas formula (Shukla et al., 1996).
- A given baseline is recomputed if this probability exceeds a threshold

Experiments

Parameters:

- For bootstrapping: 100 replications with samples of 1% of relevant records.
- Intervals built on the basis of 3 standard deviations.
- Anomalies threshold was set to 0.1%.

Case 1: A Black Day for Markets

- October 10th, 2014: NASDAQ Composite Index plummeted by 2.33%
- *S-DW* contained data from 4/Jan/1965 to 10/Oct/2014 at 13:29 GMT (1,974,462 rows).
- *D-DW* contained data for 10/Oct/2014 between 13:30 and 13:35 GMT (854 rows).

	Input		Results		
	Input Facts	Coordinate groups	Output Cells	Time	Storage (est.)
European Health-Care Companies	18,623	10	50	8 min	9 KB
US Health-Care Companies	152,063	80	400	56 min	74 KB
Semiconductors firms in NASDAQ by Year	34,868	406	2030	9 min	378 KB
Water Supply firms by Year	3,518	20	100	1 min	19 KB
TOTALS	209,072	516	2,580	74 min	480 KB

Computation time is more sensitive to the number of input facts than to the number of output cells.

Experiments (cont.)

- 90 out of the 854 facts present in the Real-Time fact table were relevant.
- They demanded 450 comparisons (5 measures).
- All of them were assessed in about **627 seconds**, which represents an average of **1.39 seconds/measure/fact**.
- One of the baselines, “Semiconductors firms in NASDAQ by Year”, detected **6 anomalies.**
- As the threshold of 0.1% we had set was exceeded at baseline level (6 out of 90), at baseline cell level (1 out 1 in 6 cells) and at general level (6 out of 450), alerts were issued in the three of them.
- Ex-post analysis:
  - Five minutes after the alert, the price kept on falling for some stocks (e.g. TXN)
  - For another stock, we see that the price at the end of the day turned out to be higher (e.g. MCHP).

Experiments (cont.)

Case 2: An Apparently Quiet Day

- November 13, 2014 has been apparently a quiet day for NASDAQ market as a whole. NASDAQ composite showed an overall slight increase of almost 0.11%.
- S-DW had data from *4/Jan/1965* and *13/Nov/2014* at 13:29 GMT (3,221,378 rows).
- D-DW had data for *13/Nov/2014* between 13:30 and 14:34 GMT (1386 rows).

- Compared to Case 1, the number of input facts increased approximately a 62% and so did the baseline computation time.
- Only 110 out of 1386 facts were relevant, shielding 550 comparisons.
- All of them were assessed in 384 **seconds**, representing an average of **0.7 seconds/measure/fact**, which is lower than the figure obtained in Case 1.
- <u>No anomalies</u> were detected in any of the four baselines.

In Conclusion

- Our approach leverages a specific real-time data warehouse architecture.
- It is analyst tailored.
- It is made up by an offline phase and an online phase.
- We implemented the approach and illustrated its interest in the domain of technical analysis of stock markets.
- As future work, we will first address the optimisation of baseline computation, which might be seen as the bottleneck of our approach.
- We will particularly study strategies for an iterative computation of baselines, using a combination of application logic and database features.
- Test our approach in a more realistic data warehouse situation, where anomaly detection competes with regular analytical queries.

Merci! Avez-vous des questions?