TECHNIQUES FOR OPTIMIZING THE RELATIONSHIP BETWEEN DATA STORAGE SPACE AND DATA RETRIEVAL TIME FOR LARGE DATABASES

Virgil Chichernea[1]
Dragos-Paul Pop[2]

ABSTRACT

There are large amounts of data generated in the information society every day; this data comes from various sources, like measuring devices, public administration, mass media, telephony, GPS, the movie industry, television stations, etc. Gartner, the author of the "hype cycle" concept, defines the "BIG Data" (BIGD) concept as "high volume, velocity and/or variety information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight, decision making, and process automation". It is the single most hyped term in the market today. BIGD has drawn the attention of the IT and marketing research communities, with concrete results about "BIGD solutions" and significant resources being allocated for "BIGD projects". The objective of the hype cycle for BIGD is to help the decision makers to work with this concept in order to develop future activities in the context of the fundamental change of the cost-benefit equation terms. Optimizing the relationship between storage space and retrieval time for data stored as BIGD is a major challenge for research in the field.

This paper shows techniques for data structuring in BIGD, techniques that optimize the relationship between storage space and retrieval time under the aspect of total cost, techniques based on Boolean algebra and atom files.

Keywords: Big Data, Hype Cycle, Volume, Velocity, Variety, and Veracity, Boolean algebra, Atoms File

1. INTRODUCTION

"Big Data" is a concept coined to name the collection of data which contains an impressive volume of data, of the most diverse types, from numeric data, to text data, sounds and images, to name just a few of the data types generated daily in the world. The complexity of the "Big data" concept can be detailed by four characteristics: Volume, Velocity, Variety, and Veracity.

The volume of data spread in BIGD is impressive for data storage alone. The information society is working daily with volumes of data way bigger than terabytes, even petabytes.

1 Proffessor, Ph.D., Romanian-American University, Bucharest, Romania, chichernea.virgil@profesor.rau.ro
2 Teaching Assistant, Romanian-American University, Bucharest, Romania, Ph.D. Student, Academy of Economic Studies, Bucharest, Romania, pop.dragos.paul@profesor.rau.ro

The retrieval speed of data stored in BIGD and the transmission speed to the final user, must guarantee data access in time for decision making.

Data diversity in BIGD is a new dimension in the complexity of this concept. Usually, data stored in BIGD is structured data, but mostly unstructured data like text, sound, images, video, etc.

Reliability of data stored in BIGD is measured in the degree of trust the end user has for data obtained from these collections and it is shown that a number of about 30% of the factors in the decision making process have big doubts or even question the reliability of this data.

By Big Data analysis we define the process of examining very big quantities of data of different types, mostly unstructured, (text, sound, images) obtained at different time periods, with the purpose of identifying the properties, the correlations and the useful information contained in these data collections. The primary purpose of BIGD analysis is to help companies in their decision making processes by identifying data obtained in time, data that cannot be supplied to decision makers by conventional business intelligence (BI) tools. These new data sources may include Web server logs and Internet clickstream data, social media activity reports, mobile-phone call detail records and information captured by sensors.

Apache Hadoop is an open source software product aimed for distributed processing of BIGD records by means of networks and client-server clusters. This software product, that manages BIGD records, is equipped with functions for storing documents / records and functions for retrieving data from the stored documents. For rapid retrieval of data, the software partitions the records from BIGD on different levels and search for the data needed is done on these resulting trees.

Gartner, the author of the “hype cycle” concept, defines the “BIG Data” (BIGD) concept as "high volume, velocity and/or variety information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight, decision making, and process automation". It is the single most hyped term in the market today. BIGD has drawn the attention of the IT and marketing research communities, with concrete results about “BIGD solutions“ and significant resources being allocated for “BIGD projects”. The objective of the hype cycle for BIGD is to help the decision makers to work with this concept in order to develop future activities in the context of the fundamental change of the cost-benefit equation terms.

According to the “hype cycle” estimates, by the end of 2012 practical and productive solutions will be obtained in BIGD IT projects in the following domains: intelligent electronic devices, Supply Chain Analytics, Social Media Monitors, Speech Recognition, Web Analytics, Column Store DBMS, Predictive Analytics.

The era of databases in which large structured data collections are stored and the era of current RDBMSs is slowly starting to fade away, leaving way for BIG DATA, i.e. new

concepts used in describing and managing the exponential growth, the availability and usage of data, both structured but mainly unstructured. Big corporations and IT leaders invest a lot of resources into projects for efficient structuring of data into BIGD, structures that should ensure the needed performance on all four dimensions of BIGD: Volume , Velocity, Variety, and Veracity. In this context, we are looking to bring a small contribution to the scientific research in this field.

2. NOTIONS, CONCEPTS AND DEFINITIONS

Annotations:

- $\{R\}$ – the collection of records from Big Data (BIGD);
- $\{C\}$ – the set of addresses form the storage environment where the records from R have been stored;
- $\{S\}$ – the set of requests for data from BIGD.

Any structuring of the R set over S so that any data request $(c_i \in C)$ can be satisfied is called a BIGD structure.

For any BIGD structuring of the following type:

$$O(\{R\},\{C\},\{S\})$$

We associate for any $c_i \in C$ element a retrieval time $t(c_i)$.

The structuring of BIGD with a minimum time of retrieval for all elements $c_i \in C$ and a minimum number of addresses S is a primary objective for BIGD.

The main objective of BIGD structuring of data is the optimization of the relation between storage space S – retrieval time $t(c_i)$, under the aspect of total cost.

In mathematical terms this objective can be formulated as follows:

In the given set $\mathcal{M}\{O(\{R\},\{C\},\{S\})\}$, identify the structure O that optimizes the relationship described below:

$$min\ t(c_i)\ ,\forall\ c_i \in C \text{ for } min\ Cost\big(O(\{R\},\{C\},\{S\})\big) \text{ (uder the aspect of total cost)} \quad (2.1)$$

A general solution for this problem, a general goal pursued by software products that manage data and information from many fields, like: bibliographic documentation, medical computer science, administration, spatial programs, etc., is hard to obtain because of the complex structures of the $\{R\},\{C\},\{S\}$ sets.

For the completion of this goal there have been developed a large array of techniques for data organization going from files to databases and from the simplest to the most complex RDBMSs.

3. BOOLEAN ALGEBRA FOR RECORD SETS

In order to obtain an efficient technique for organizing BIGD in the conditions of (2.1), let's make the following technical considerations.

Let there be a large BIGD database that contains the record set (object set) $\{R\}$ and let's consider a number of keywords (fields contained in the BIGD records), denoted as $k_1, k_2, \dots, k_n$, with the property that any record from BIGD contains at least one keyword.

Annotations:

- $R(k_i)$ – the set of records that contain the keyword k_i,
- $A(k_i)$ – the list of addresses for the records in this set

Using these annotations, the set of records from BIGD can be partitioned in classes: $\mathcal{R}(k_i)$.

Let there be $\mathcal{R} = \{R(k_i)\,;\mathrm{i} = 1,2,\dots,\mathrm{n}\}$ where the operations $\cap,\cup,-$ have the following meanings:

$$R(k_1) \cap R(k_2) = R(k_1 \wedge k_2)$$

$$R(k_1) \cup R(k_2) = R(k_1 \vee k_2)$$

$$\bar{R}(k_1) \cap \bar{R}(k_2) = \bar{R}(k_1 \wedge k_2)$$

Definition 1: O family of parts P of a set $\mathcal{M}$ is a Boolean algebra if and only if:

$$A \in P \; and \; B \in P \; \rightarrow A \cup B \in P \; si \; \bar{A} \in P, \bar{B} \in P$$

The set $\mathcal{R} = \{R(k_i)\,;\mathrm{i} = 1,2,\dots,\mathrm{n}\}$ generates a Boolean algebra which we denote by $\mathcal{B}(R)$.

We observe that, by using these annotations, any data request $(c_i \in C)$ from BIGD can be written as a Boolean function of keywords like:

$$f(k_1, k_2, \dots, k_n) = \bigvee_{i \le n}(\bigwedge_{i \le n} k_i)$$

The answer to this data request is contained in a collection of records $B \in \mathcal{B}(\mathcal{R})$, so the set $\mathcal{B}(\mathcal{R})$ can be called the Boolean algebra of all possible answers.

Further, let's search to identify in the Boolean algebra $\mathcal{B}(\mathcal{R})$ a set of elements that have the property that two by two are disjoint. They can be easily be highlighted if we account for the total number of sets of the following type:

$$C_j = \bigcap_{i=1}^{n} \widetilde{R}_j(k_i) \qquad (3.1)$$

Where $\widetilde{R}_j(k_i)$ is $R_j(k_i)$ or $\overline{R}_j(k_i)$.

This way, we can build the set of 2^nelements denoted by:

$$C_1, C_2, \dots, C_m, C_{m+1}, \dots, C_{2^n} \qquad (3.2)$$

of which some can be empty.

Concerning the nonempty elements $C_1, C_2, \dots, C_m$ the following lemmas can be easily proved:

Lemma 1: The elements C_j and C_k are distinct for any $j \neq k$; meaning $C_j \cap C_k = \emptyset$ for any $j \neq k$.

Lemma 2: For any $B \in \mathcal{B}(\mathcal{R})$ we have $B \cap C_j = \begin{cases} C_j \\ \emptyset \end{cases}$, for any j.

In this case, the nonempty elements C_j defined by (3.1) will be named as atoms.

Lemma 3: Any subset B of records $R(k_i)$ from BIGD is a union of atoms C_i.

To prove this lemma, we notice that from (3.1) it follows that:

$$BIGD = \bigcup_{i=1}^{n} R(k_i) = \bigcup_{i=1}^{n} C_i$$

$$B = B \cap BIGD = \bigcap_{i=1}^{m} B \cap C_i;$$

But $B \cap C_i = \begin{cases} C_i \\ \emptyset \end{cases}$ according to lemma 2, so lemma 3 is proved.

4. ORGANIZING BIGD ON THE BASIS OF THE BOOLEAN ALGEBRA OF ALL POSSIBLE ANSWERS

For the achievement of an efficient technique for organizing BIGD let's consider that instead of organizing BIGD in the following form:

$$O\big((\{R\}, \{A(k_i)\}), \{Q\}, \{S\}\big) \qquad (4.1)$$

which makes up the classical structure on which most RDBMSs operate on, the following organization:

$$O\big((\{R\}, \{A(C_i)\}), \{Q\}, \{S\}\big) \qquad (4.2)$$

in which the set $\{A(C_i)\}$ is made up of the lists of addresses of the records that contain the atoms C_i.

Such an organization has the following advantages:

a) Any address of records contained in BIGD appears on only one of the lists $\{A(C_i)\}$, in other words: the number of addresses $\{A(C_i); i = 1,2, \dots, m\} \leq$ the number of addresses $\{A(k_i); i = 1,2, \dots, n\}$, because $C_i \cap C_j = 0$ for $i \neq j$;
b) Any set of records $B \in BIGD$ which has to be found to satisfy a given demand $f(k_1, k_2, \dots, k_n)$ is a union of disjoint atoms (see lemma 3). So in the organization at (4.2) we never take into consideration the intersections of lists of addresses and we never eliminate the duplications in the taking of the unions of address lists, the way it happens in other known techniques [11].
c) The procedures of translations of a Boolean function $f(k_1, k_2, \dots, k_n)$ of random keywords in a union of atoms are extremely simple.

To prove this it is sufficient to consider that any Boolean function $f(k_1, k_2, \dots, k_n)$ can always be expressed in a normal form of disjunctive clauses, each clause being a conjunction of keywords k_i or $\bar{k}_j$, like this:

$$f(k_1, k_2, \dots, k_n) = \left(k_1 \wedge k_2 \wedge \bar{k}_4\right) \vee \left(k_2 \wedge \bar{k}_3 \wedge k_4\right)$$

Definition 2: A disjunctive normal form of a Boolean function is a developed disjunctive normal form is any variable appears once and only once in any clause, either in negative form or not (never under both forms) [11];

It is observed that the transformation from a disjunctive normal form to a developed disjunctive normal form is done very easily by replacing clause 0 from the disjunctive normal form that does not contain a key k_j with $\left(\emptyset \wedge k_j\right) \vee \left(\emptyset \wedge \bar{k}_j\right)$.

With these specifications we have:

$$f(k_1, k_2, \dots, k_n) = \bigvee_{j \leq n}\left(\bigvee_{i \leq n} k_{ji}\right) \text{ where } k_{ji} = \begin{cases} k_i \\ \bar{k}_i \end{cases}$$

$$\text{and } R(f) = R\left(\bigvee\left(\bigvee \tilde{k}_{ji}\right)\right) = \bigcup_{j \leq n} \bigcap_{i \leq n} R\left(\tilde{k}_{ji}\right) = \bigcup_{j \leq n} \bigcap_{i \leq n} \tilde{R}(k_i) \quad (4.3)$$

So, any set B that satisfies a request $f(k_1, k_2, \dots, k_n)$ is a union of atoms of the Boolean algebra $\mathcal{B}(\mathcal{R})$.

From the considerations listed at points a, b, c it follows that:

Theorem 1: The organization of BIGD under the form at (4.3) is better than the form at (4.2) viewed in terms of the relationship between storage space – retrieval time under the conditions of a minimum total cost.

5. PROCEDURES FOR GENERATING ATOMIC BOOLEAN ALGEBRA

Method I:

To simplify the exposure, let's consider that BIGD contains 12 documents (records) and 4 keywords. In the following we will refer only to the addresses of the records (documents) and not to their contents. Let's denote the keywords taken into consideration with k_1, k_2, k_3, k_4 and the BIGD records with 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12.

Suppose that $R(k_i); i = 1, 2, 3, 4$ are as follows:

$$R(k_1) = \{2, 4, 5, 7, 8, 10, 11, 12\}$$
$$R(k_2) = \{1, 2, 7, 10\}$$
$$R(k_3) = \{1, 4, 5, 8, 11\}$$
$$R(k_4) = \{3, 5, 6, 8, 9, 12\} \qquad (5.1)$$

To highlight the atoms C_j it is sufficient to express the sets $R(k_i)$ in the form of columns in table 1, where we will denote by the value "1" the belonging of the record from column 1 of the table to the set $R(k_i)$ and by the value "0" the not belonging of the record to the set $R(k_i)$.

Table 1

Record number	$R(k_1)$	$R(k_2)$	$R(k_3)$	$R(k_4)$
1	0	1	1	0
2	1	1	0	0
3	0	0	0	1
4	1	0	1	0
5	1	0	1	1
6	0	0	0	1
7	1	1	0	0
8	1	0	1	1
9	0	0	0	1
10	1	1	0	0
11	1	0	1	0
12	1	0	0	1

By reading the rows of table 1 we can write the atoms of the Boolean algebra, according to (3.2) like this:

$$C_1 = \bar{R}(k_1) \cap \bar{R}(k_2) \cap \bar{R}(k_3) \cap R(k_4) = \{3, 6, 9\}$$
$$C_2 = R(k_1) \cap \bar{R}(k_2) \cap R(k_3) \cap \bar{R}(k_4) = \{4, 11\}$$
$$C_3 = R(k_1) \cap \bar{R}(k_2) \cap R(k_3) \cap R(k_4) = \{5, 8\}$$

$$C_4 = R(k_1) \cap R(k_2) \cap \bar{R}(k_3) \cap \bar{R}(k_4) = \{2, 7, 10\}$$
$$C_5 = R(k_1) \cap \bar{R}(k_2) \cap \bar{R}(k_3) \cap R(k_4) = \{12\}$$
$$C_6 = \bar{R}(k_1) \cap R(k_2) \cap R(k_3) \cap \bar{R}(k_4) = \{1\} \quad (5.2)$$

The numbering of atoms is arbitrary.

Method II:

From the definition of C_j the following recurrence formula is obtained:

$$C_j = \cap_{i-1}^{n} \tilde{R}_j(k_i) = \left(\cap_{i=1}^{n-1} \tilde{R}_j(k_i)\right) \cap \tilde{R}_j(k_n) \quad (5.3)$$

From (5.3) the iterative generation method of atoms C_j follows. Let's follow this method for the example above.

Step 1.

Let's consider all the records that contain the keyword k_1; that is the set

$$R(k_1) = \{2, 4, 5, 7, 8, 10, 11, 12\}$$

Step 2.

Let's consider the sets $R(k_1) \cap \bar{R}(k_2)$; $R(k_1) \cap R(k_2)$ and $\bar{R}(k_1) \cap R(k_2)$ and we eliminate the empty sets:

$$R(k_1) \cap \bar{R}(k_2) = \{4, 5, 8, 11, 12\}$$
$$R(k_1) \cap R(k_2) = \{2, 7, 10\}$$
$$\bar{R}(k_1) \cap R(k_2) = \{1\}$$

Step 3.

We consider the not empty sets obtained at Step 2 and we intersect them with $R(k_3)$ and $\bar{R}(k_3)$ and we only keep the not empty sets. Then we also consider the set $\bar{R}(k_1) \cap \bar{R}(k_2) \cap \bar{R}(k_3) \cap R(k_4)$ and we keep it if it is not empty:

$$R(k_1) \cap \bar{R}(k_2) \cap R(k_3) = \{4, 5, 8, 11\}$$
$$R(k_1) \cap \bar{R}(k_2) \cap \bar{R}(k_3) = \{12\}$$
$$R(k_1) \cap R(k_2) \cap R(k_3) = \{0\} \; - \text{ gets eliminated}$$
$$R(k_1) \cap R(k_2) \cap \bar{R}(k_3) = \{2, 7, 10\}$$
$$\bar{R}(k_1) \cap R(k_2) \cap R(k_3) = \{1\}$$
$$\bar{R}(k_1) \cap R(k_2) \cap \bar{R}(k_3) = \{0\} \; - \text{ gets eliminated}$$
$$\bar{R}(k_1) \cap \bar{R}(k_2) \cap R(k_3) = \{0\} \; - \text{ gets eliminated}$$

Step 4.

We keep the not empty sets obtained in Step 3 and we intersect them with the sets $R(k_4)$ and $\bar{R}(k_4)$ then we also consider the set $\bar{R}(k_1) \cap \bar{R}(k_2) \cap \bar{R}(k_3) \cap R(k_4)$:

$$R(k_1) \cap \bar{R}(k_2) \cap \bar{R}(k_3) \cap R(k_4) = \{12\}$$
$$R(k_1) \cap \bar{R}(k_2) \cap \bar{R}(k_3) \cap \bar{R}(k_4) = \{0\} \ - \text{ gets eliminated}$$
$$R(k_1) \cap \bar{R}(k_2) \cap R(k_3) \cap R(k_4) = \{5, 8\}$$
$$R(k_1) \cap \bar{R}(k_2) \cap R(k_3) \cap \bar{R}(k_4) = \{4, 11\}$$
$$R(k_1) \cap R(k_2) \cap \bar{R}(k_3) \cap \bar{R}(k_4) = \{2, 7, 10\}$$
$$R(k_1) \cap R(k_2) \cap \bar{R}(k_3) \cap R(k_4) = \{0\} \ - \text{ gets eliminated}$$
$$\bar{R}(k_1) \cap R(k_2) \cap R(k_3) \cap R(k_4) = \{0\} \ - \text{ gets eliminated}$$
$$\bar{R}(k_1) \cap R(k_2) \cap R(k_3) \cap \bar{R}(k_4) = \{1\}$$

and

$$\bar{R}(k_1) \cap \bar{R}(k_2) \cap \bar{R}(k_3) \cap R(k_4) = \{3, 6, 9\}$$

We observe that we obtained the same atoms $C_j$as in Method I.

6. QUERY PROCEDURE FOR THIS STRUCTURE

In essence, this procedure aims to transpose a query into atom unions. We consider a request to BIGD written in the form of a Boolean function:

$$f(k_1, k_2, \dots, k_n) = (k_1 \wedge k_2 \wedge k_3 \wedge k_4) \vee \left(k_2 \wedge \bar{k}_3 \wedge k_4\right) \vee k_4$$

We are looking to transform this query in a developed disjunctive normal form, i.e.:

$$\begin{aligned} f(k_1, k_2, \dots, k_n) = \ & \left(k_1 \wedge k_2 \wedge k_3 \wedge \bar{k}_4\right) \vee \\ & \left(k_1 \wedge k_2 \wedge \bar{k}_3 \wedge \bar{k}_4\right) \vee \\ & \left(k_1 \wedge k_2 \wedge \bar{k}_3 \wedge k_4\right) \vee \\ & \left(\bar{k}_1 \wedge k_2 \wedge \bar{k}_3 \wedge k_4\right) \vee \\ & \left(\bar{k}_1 \wedge \bar{k}_2 \wedge \bar{k}_3 \wedge k_4\right) \end{aligned} \quad (5.4)$$

We search for the correspondence between each term of the disjunctive normal form and generated atoms C_i.

We observe that from the five terms of the query only the $\left(k_1 \wedge k_2 \wedge \bar{k}_3 \wedge \bar{k}_4\right)$ corresponds to the atom $C_4 = R(k_1) \cap R(k_2) \cap \bar{R}(k_3) \cap \bar{R}(k_4)$ and the $\left(\bar{k}_1 \wedge \bar{k}_2 \wedge \bar{k}_3 \wedge k_4\right)$ corresponds to the atom $C_1 = \bar{R}(k_1) \cap \bar{R}(k_2) \cap \bar{R}(k_3) \cap R(k_4)$. It follows that the set of records from BIGD that satisfies the query (5.4) is made up of the records found at the addresses $\{2, 7, 10\}$ and $\{3, 6, 9\}$.

7. THE UPDATE PROCEDURE

Concerning the update process, for this technique of BIGD organization we distinguish the following aspects:

1. The change of record values;
2. The erase of records form BIGD;
3. The addition of new records to BIGD;
4. The change of keywords.

We notice that in the update process, like in the query process for BIGD, in the first stage we must select the affected records. For this step of selecting records we will use the specified query function and then we will perform the specific operations of the update process.

For the aspects described at points 1 and 2, through the query function we will select the set of records affected by the query and then we will perform the update operations on this set of records. For the addition of new records to BIGD we will use the procedure used at the generation step.

Modifying the keywords implies the following aspects:

a) Adding new keywords;
b) Erasing keywords.

Consider the atoms:

$$C_j = \bigcap_{i=1}^{n} \tilde{R}_j(k_i)$$

where $\tilde{R}_j(k_i)$ is $R(k_i)$ or $\bar{R}(k_i)$ obtained for the keywords $k_1, k_2, k_3, \dots k_n$. If we add to the existing *n* keywords another *m* keywords, the it will be necessary to undergo another *m* iterations after Method II. The starting point of the iterative method is the atoms C_j obtained for the *n* keywords.

Erasing keywords is done through the same procedure but in reverse. By eliminating the keyword k_{n+1} from the set $C_j = \bigcap_{i=1}^{n+1} \tilde{R}_j(k_i)$ the set $C_j = (\bigcap_{i=1}^{n} \bar{R}(k_i)) \cap R(k_{n+1})$ disappears and on the set $C_1 = \bigcap_{i=1}^{n} \tilde{R}_j(k_i)$ a removal operation is made to eliminate the duplicates with the union of the address lists contained in these duplicate sets.

8. ACKNOWLEDGEMENT

This work was co-financed from the European Social Fund through Sectorial Operational Program Human Resources Development 2007-2013, projects POSDRU/107/1.5/S/77213 and POSDRU/88/1.2/S/55287 „Ph.D. for a career in interdisciplinary economic research at the European standards"

REFERENCES

1. *Big data* - Wikipedia, the free encyclopedia, *en.wikipedia.org/wiki/Big-data*

2. IBM What is *big data*? - Bringing *big data* to the enterprise, *www.ibm.com/software/data/bigdata/*
3. *Big Data* – What Is It? | SAS, *www.sas.com/big-data/index.html*
4. *Big data*: The next frontier for innovation, competition, and productivity, www.mckinsey.com/.../big_data_the_next_frontier_for_innovation
5. *Big Data Architecture bigdataarchitecture.com/*
6. 6. What is *big data analytics*? - Definition from WhatIs.com, *searchbusinessanalytics.techtarget.com/.../big-data-analytics*
7. 7. IBM - What is *Hadoop* – Bring the power of *Hadoop* to the enterprise , *www.ibm.com/software/data/infosphere/hadoop/*
8. *Big Data* | Microsoft SQL Server, *www.microsoft.com/sqlserver/en/us/.../big-data.aspx - United States*
9. Big Data Analytics Hadoop | mapr.com. www.mapr.com/Free-Downloaad
10. *Hype Cycle for Big Data, 2012* - Hadoop & Big Data – Consultant, www.hadoopconsultant.nl/.../hype_cycle_for_big_data_2012_23504...
11. *Chichernea V. – Large databse organization technics,* Studii si cercetari de matematica, *April 1977*
12. *BOOLEAN ALGEBRA,* www.cimt.plymouth.ac.uk/projects/mepres/alevel/discrete_ch11.pdf, www.informationoptimized.com.

Source: http://www.economist.com
Source: http://blog.twitter.com
Source: http://newsroom.fb.com/