ACADEMY OF ECONOMIC STUDIES

ISSUE
2

Database Systems Journal

ISSN: 2069 – 3230 Volume I (December 2010)

Journal edited by Economic Informatics Department

DBJOURNAL BOARD

Contact
Calea Dorobanților, no. 15-17, room 2017, Bucharest, Romania
Web: http://dbjournal.ro/
E-mail: editor@dbjournal.ro

Contents

Column-Oriented Databases, an Alternative for Analytical Environment

Gheorghe MATEI
Romanian Commercial Bank, Bucharest, ROMANIA
George.matei@bcr.ro

It is widely accepted that a data warehouse is the central place of a Business Intelligence system. It stores all data that is relevant for the company, data that is acquired both from internal and external sources. Such a repository stores data from more years than a transactional system can do, and offer valuable information to its users to make the best decisions, based on accurate and reliable data. As the volume of data stored in an enterprise data warehouse becomes larger and larger, new approaches are needed to make the analytical system more efficient. This paper presents column-oriented databases, which are considered an element of the new generation of DBMS technology. The paper emphasizes the need and the advantages of these databases for an analytical environment and make a short presentation of two of the DBMS built in a columnar approach.

***Keywords**: column-oriented database, row-oriented database, data warehouse, Business Intelligence, symmetric multiprocessing, massively parallel processing.*

1 Introduction

In the evolution of computing science, three generations of database technology are identified since the 60's till nowadays. The first generation started in the 60's and its main purpose was to enable disparate but related application to share data otherwise than passing files between them.

The publishing of "*A Relational Model of Data for Large Shared Data Banks*" by E. F. Codd marked the beginning of the second generation of DBMS (*database management systems*) technology. Codd's premise was that data had to be managed in structures developed according to the mathematical set theory. He stated that data had to be organized into tuples, as attributes and relations.

A third generation began to emerge in the late 90's and now is going to replace second-generation products. Multi-core processors became common, 64-bit technology is used largely for database servers, memory is cheaper and disks are cheaper and faster than ever before.

A recent IDC study [1] examines emerging trends in DBMS technology as elements of the third generation of such technology. It considers that, at the current rate of development and adoption, the following innovations will be achieved in the next five years:

- most data warehouses will be stored in a columnar fashion;
- most OLTP (*On-Line Transaction Processing*) databases will either be augmented by an in-memory database or reside entirely in memory;
- most large-scale database servers will achieve horizontal scalability through clustering;
- many data collection and reporting problems will be solved with databases that will have no formal schema at all.

This study examines how some innovations in database technology field are implemented more and more. Most of these technologies have been developed for at least ten years, but they are only now becoming widely adopted.

As Carl Olofson, research vice president for database management and data integration software research at IDC, said, "*many of these new systems encourage you to forget disk-based partitioning schemes, indexing strategies and buffer management, and embrace a world of large-memory models, many processors with many cores,*

clustered servers, and highly compressed columnwise storage".

From the innovations that the study considers that will be achieved in the next years, this paper presents the columnar data storage.

2. The need for column-oriented databases

The volume of data acquired into an organization is growing rapidly. So does the number of users who need to access and analyse this data. IT systems are used more and more intensive, in order to answer more numerous and complex demands needed to make critical business decisions. Data analysis and business reporting need more and more resources. Therefore, better, faster and more effective alternatives have to be found. Business Intelligence (BI) systems are proper solutions for solving the problems above. Decision-makers need a better access to information, in order to make accurate and fast decisions in a permanent changing environment. As part of a BI system, reporting has become critical for a company's business.

Years ago, reports prepared by analysts were addressed only to the company's executive management. Nowadays, reporting has become an instrument addressed to decision-makers on all organizational levels, aiming to improve the company's activity, to ensure decision quality, control costs and prevent losses.

As already mentioned, the volume of data acquired into a company is growing permanently, because business operations expand and, on the other hand, the company has to interact with more sources of data and keep more data online. More than ever before, users need a faster and more convenient access to historical data for analysing purposes. Enterprise data warehouses are a necessity for the companies that want to stay competitive and successful. More and more reports and ad-hoc queries are requested to support the decision making process. At the same time, companies have to run audit reports on their operational and historical data in order to ensure compliance [2].

These new demands add more pressures upon IT departments. More and more hardware resources are needed in order to store and manage an increasing volume of data. The increasing number of queries needs larger amounts of CPU cycles, so more processors, having a higher performance, must be added to the system

The size of the data warehouses storing this data is increasing permanently, becoming larger and larger. While five years ago the largest data warehouses were around 100 terabytes in size, now a data warehouse size at the petabyte level is no longer unusual. The challenge is to maintain the performance of these repositories, which are built, mostly, as relational structures, storing data in a row-oriented manner. The relational model is a flexible one and it has proven its capacity to support both transactional and analytical processing. But, as the size and complexity of data warehouses have increased, a new approach was proposed as an alternative on the row-oriented approach, namely storing data in a column-oriented manner. Unlike the row-oriented approach, where the data storage layer contains records (rows), in a column-oriented system it contains columns. This is a simple model, more adequate for data repositories used by analytical applications, with a wide range of users and query types.

Researches indicate that the size of the largest data warehouse doubles every three years. Growth rates of system hardware performance are being overrun by the need for analytical performance [3]. The volume of data needed to be stored is growing due to more and various requirements for reporting and analytics, from more and more business areas, increased time periods for data retention, a greater number of observations loaded in data warehouses and a greater number of attributes for each observation. This is true if taking into consideration only structured data. But nowadays, organizations collect a larger and larger volume of unstructured data, as images,

audio and video files, which need a much greater storing space than structured data.

Row-oriented databases have been designed for transactional processing. For example, in the account management system of a bank, all attributes of an account are stored in a single row. Such an approach is not optimal in an analytical system, where a lot of read operations are executed in order to access a small number of attributes from a vast volume of data. In a row-oriented architecture, system performance, users' access and data storage become major issues very quickly [4]. As they are designed to retrieve all elements from several rows, row-oriented databases are not well suited for large scale processing, as needed in an analytical environment. As opposed to transactional queries, analytical queries typically scan all the database's records, but process only a few elements of them. In a column-oriented database all instances of a single data element, such as account number, are stored together so they can be accessed as a unit. Therefore, column-oriented databases are more efficient in an analytical environment, where queries need to read all instances of a small number of data elements.

System performance enhances spectacularly in a column-oriented solution, because queries search only few attributes, and they will not scan the attributes that are irrelevant for those queries. Requested data is found faster, because less sort operations have to be performed.

A typical feature of evolved BI systems is their capability to make strategic business analyses, to process complex events and to drill deeply into data. As the volume of data becomes impressive and performance demands required by users are likely to outpace, it is obviously that row-oriented relational database management systems stopped to be the solution for implementing a BI system having powerful analytical and predictive capabilities. A new model tends to come into prominence as an alternative on developing analytical databases, namely one that manages data by columns.

A column-oriented DBMS stores data in a columnar manner and not by rows, as classic DBMS do. In the columnar approach, each attribute is stored in a separate table, so successive values of that attribute are stored consecutively. This is an important advantage for data warehouses where, generally, information is obtained by aggregating a vast volume of data. Therefore, operations as MIN, MAX, SUM, COUNT, AVG and so forth are performed very quickly [5].

When the tables of a database are designed, their columns are established. The number of rows will be determined when the tables will be populated with data. In a row-oriented database, data is stored in a tabular manner. The data items of a row are stored one after another; rows are also stored one after another, so the last item of a row is followed by the first item of the next row.

In a column-oriented database, the data items of a column are stored one after another, and also are the columns; so the last item of a column is followed by the first item of the next column.

3. Differences between the row-oriented and column-oriented approaches

In a typical relational DBMS, data is stored and managed as rows, each row containing all the attributes of an element of that entity (table). Such systems are used by transactional applications which, at a certain moment, generate or modify one or a small number of records. Unlike transactional applications, which use all, or almost all the attributes of a record, analytical and BI applications scan few attributes (columns) of a vast number of records. Most often, they have to aggregate data stored in those columns in order to meet the users' demands. Because of the row-oriented structure of the database, the entire record has to be read in order to access the required attributes. This fact causes the reading of a vast amount of unuseful additional data in order to access the requested information.

Figure 1 shows that much more data than needed is read to satisfy the request for the total volume of term deposits opened at the branches in Bucharest.

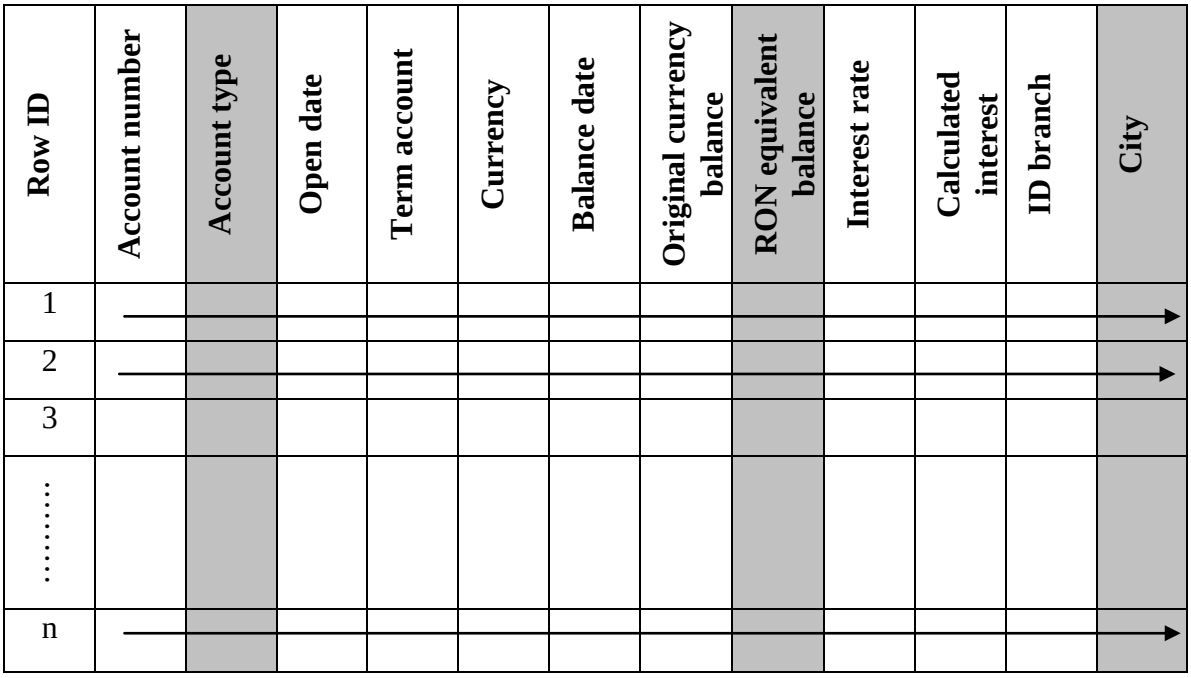

Fig. 1. Analytical request in a row-oriented database

Row-oriented databases were designed for transactional applications, and they are optimized for retrieval and processing of small data sets. Seldom, to support analytical requests, it is necessary to build additional indexes, pre-aggregating data structures, or special materialized views and cubes. All these aspects require additional processing time and data storage. However, because they are built in order to provide quickly results for queries that were known at the design stage, they will not have the same performance when ad-hoc queries, that were not foreseen before, are performed.

The business demands require the storage of many data items. But any user wants to get information as soon as possible. Therefore, a proper solution for data organization has to be implemented in order to ensure a good performance of the system.

Several technical solutions can be used to improve system performance, such as partitioning, star indexes, query pre-processing, bitmap and index joins, or hashing. These solutions aim to offer support for more specific data retrieval, but they still have to examine the entire content of a row.

Taking into consideration those presented above, a new approach was proposed, to store data along columns. In such data organization, each column is stored separately and the system selects only the columns requested by users. In every column data is stored in row order, so that the 50^{th} entry for each column belongs to the same row, namely the 50^{th} row.

Figure 2 shows that the same query as those in figure 1 reads less data in a column-oriented system, in order to provide the same result. No additional indexes have to be built for improving query performance, because every column forms an index. This fact reduces the number of I/O operations and enables quick access to data, without the need to read the entire database. Data from each column is stored contiguously on disk. Column values are joined into rows based on their relative position in each column. As a result of the column-oriented architecture, only those columns needed for a specific query are read form disk. Because in an

analytical environment most of queries need to retrieve only few columns, this vertical partitioning approach produces important I/O savings. This fact contributes to system performance improvement, as regards the query execution time.

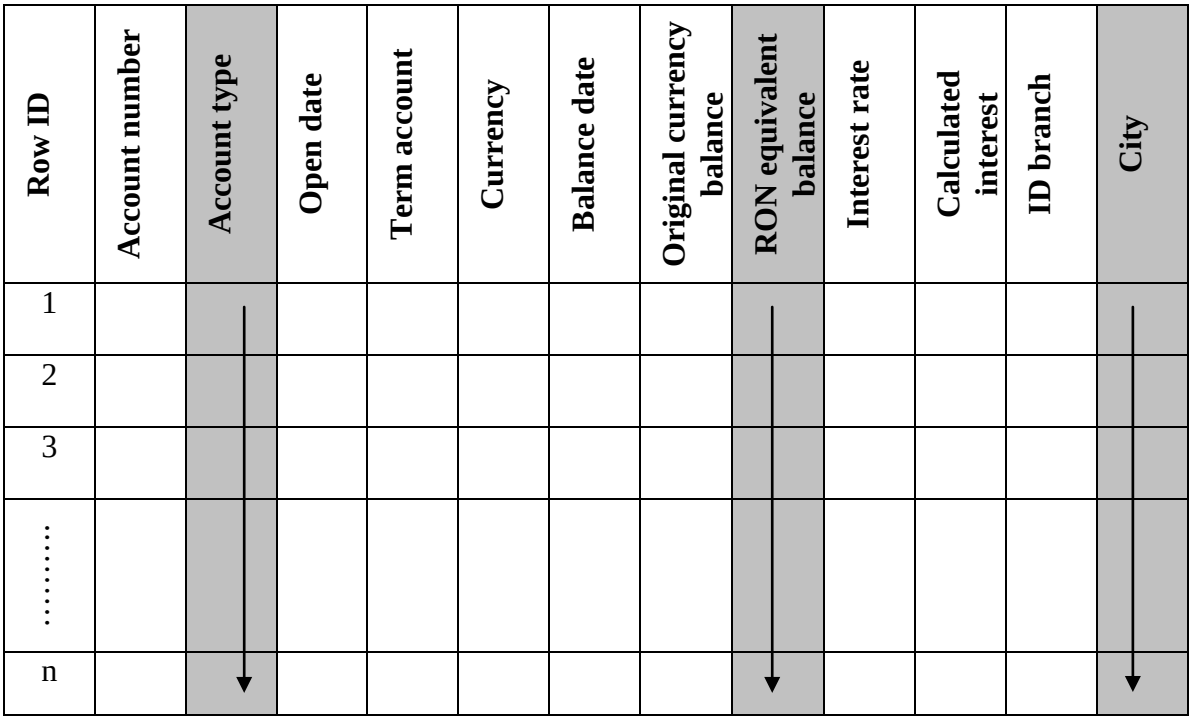

Fig. 2. Analytical request in a column-oriented database

Note. Figure 2 presents a reunion of the tables in a column-oriented database. In fact, each table has two columns: one containing the row ID, and the other, the values of the appropriate attribute. Because of the limited space on the page, the row ID column is not multiplied for every table, and the attribute columns are close together.

Comparing the two figures above, it's easy to observe that the same request has to read more data in a row-oriented structure than in a column-oriented one. In order to read a certain attribute in a row-oriented structure, all the adjacent attributes have to be read, even if they are not interesting for the requester. In a column-oriented structure, since all values of an attribute are stored together, consecutively, this problem doesn't exist [6].

A column-oriented database is faster than a row-oriented one, because its processing is not affected by unnecessary content of rows. As long as many database tables can have dozens of columns and most business requests need only a few of them, the columnar approach is a proper solution for analytical systems.

Talking about the efficiency of a column-oriented system, some remarks are to be made concerning processing time. Thus, such a system is more efficient when it's necessary to aggregate a large number of rows, but only a small number of columns are requested. If many columns of a small number of rows have to be processed, a row-oriented system is more efficient than a column-oriented one. The efficiency is even greater when row size is relatively small, because the entire row can be retrieve with a single disk seek.

Updating an entire column at once is faster in a column-oriented database. All the data of that column is modified through only one updating command, without the need to read all columns of each row. But writing or updating a single row is more efficient in a row-oriented database if all attributes are supplied at the same time, because the entire

row can be written with a single disk access, whereas writing to multiple columns requires multiple writes.

SQL queries in a column-oriented database are identical with those in a row-oriented database, without any modification. What is different, is the way that the database administrator has to think about data. While in a row-oriented database he thinks in terms of individual transactions, in a column-oriented database he has to think in terms of similar items derived from sets of transactions. From the indexing point of view, he has to pay more attention to the cardinality of the data, because an index is related with a subject, such as the balance account, and not with an entire transaction with all its fields.

4. Advantages of the column-oriented approach

Column-oriented databases provide important advantages towards the row-oriented ones, some of them being presented below.

Column-oriented databases provide a **better performance** for analytical requests. In the row-oriented approach, the system performance decreases significantly as the number of simultaneous queries increases. Building additional indexes in order to accelerate queries becomes uneffective with a large number of diverse queries, because more storage and CPU time are required to load and maintain those indexes. In a column-oriented system indexes are built to store data, while in a row-oriented system they represent the way to point to the storage area that contains the row data. As a result, a column-oriented system will read only the columns required in a certain query.

On the other hand, as they store data as blocks by columns rather than by rows, the actions performed on a column can be completed with less I/O operations. Only those attributes requested by users are read from disk. Although a row-oriented table can be partitioned vertically, or an index can be created for every column so it could be accessed independently, the performance is significantly lower than in a column-oriented structure [7]. And taking into consideration that I/O operations are the bottleneck of a database application, the column-oriented approach proves its superiority against the row-oriented one.

Unlike the row-oriented approach, the column-oriented approach allows **rapid joins and aggregations**. Tables are already sorted, so there is no need to sort them before merge or join them. In addition, accessing data along columns allows incremental data aggregation, which is very important for BI applications. In addition, this approach allows parallel data access, improving the system performance. Thereby, complex aggregations can be fulfiled by the parallel processing of columns and then joining data in order to achieve the final result.

Column-oriented databases need a **smaller disk space** to store data than row-oriented databases. To accommodate the sustained increase of volume of data, additional structures – as indexes, tables for pre-aggregation and materialized views, are built in row-oriented systems. Column-oriented databases are more efficient structures. They don’t need additional storage for indexes, because data is stored within the indexes themselves. Bitmap indexes are used to optimize data store and its fast retrieval. That’s why in a column-oriented database queries are more efficient than in a row-oriented one.

Moreover, a higher data compression rate can be achieved in a column-oriented database than in a row-oriented one. It is well known that compression is more effective when repeated values are presented, and values within a column are quite similar to each other. A column-oriented approach allows the ability to highly compress the data due to the high potential for the existence of similar values in adjacent rows of a certain column. In a row-oriented database, values in a row of a table are not likely to be very similar; therefore, they cannot be compressed as efficient as in a column-oriented database.

Concerning the repository presented in figures 1 and 2, there is no doubt that many repeated values will be found within the CITY column, but no repetition will be found between CITY and another attribute in a row.

Data loading is a **faster process** if it's executed in a column-oriented database than in a row-oriented one. As known, to load data in a data warehouse involves to perform more activities. Data is extracted from source systems and loaded into a staging area. This is the place where data transformations and joins are performed in order to denormalize data and load it into the data warehouse as fact and dimension tables. Then the needed indexes and views are created. In a row-oriented structure, all data in a row (record) is stored together, and indexes are built taking into consideration all the rows. In a column-oriented structure, data of each column is stored together and the system allows the parallel loading of the columns, ensuring a shorter time for data loading.

Taking into consideration the features presented above, it can be stated that a column-oriented database is a scalable environment that keeps providing fast queries when the volume of data, the number of users and the number of simultaneous queries are increasing.

But this thing doesn't mean that all repositories have to be built in a columnar manner. A column-oriented architecture is more suitable for data warehousing, with selective access to a small number of columns, while a row-oriented one is a better solution for OLTP systems. For an OLTP system, which is heavily loaded with interactive transactions, a row-oriented architecture is well-suited. All data for a certain row is stored as a block. In such an architecture, all the attributes of a record are written on disk with a single command, this thing ensuring a high performance for writing operations. Usually, an operation in such a system creates, queries or changes an entry in one or more tables. For an OLAP (*On-Line Analytical Processing*) system, designed for analytical purposes, which involve processing of a large number of values of few columns, a column-oriented architecture is a better solution. A data warehouse, which is the central place of an analytical system, must be optimized for reading data. In such an architecture, data for a certain column is stored as a block, so the analytical queries, which usually aggregate data along columns, are performed faster. A column-oriented system reads only the columns required for processing a certain query, without bringing into memory irrelevant attributes. Such an approach provides important advantages concerning the system performance, because typical queries involve aggregation of large volumes of data [8].

5. Examples of column-oriented database systems

Besides the column-oriented approach, another important innovation applied in data warehousing consists in the way in which data is processed. Two major techniques are used to design a data warehouse architecture: *symmetric multiprocessing* (SMP) and *massively parallel processing* (MPP). It couldn't be certified the superiority of one approach against the other. Each of these solutions has its own supporters, because both of them are valid approaches and, when properly applied, lead to notable results.

Two database systems are presented in the next sections, each of them using one of the two types of architecture.

5.1.Sybase IQ

Sybase IQ is a high-performance decision support server designed specifically for data warehousing. It is a column-oriented relational database that was built, from the very beginning, for analytics and BI applications, in order to assist reporting and decision support systems. This fact offers it several advantages within a data warehousing environment, including performance, scalability and cost of ownership benefits.

A Sybase IQ database is different from a conventional relational database, because its main purpose is to allow data analysis and not its writing or updating. While in a conventional database the most important thing is to allow many users to update the database instantly and accurately, without interfering with one another, in a Sybase IQ database the most important thing is to ensure fast query response for many users.

Sybase IQ has a column-oriented structure and has its own indexing technology that ensures high performance to reporting and analytical queries, which are performed, as its developers state, up to 100 times faster than in a traditional DBMS. Sybase IQ offers enhanced features for data loading. Its flexible architecture ensures that the system will provide rapidly the requested information, within seconds or minutes at most, no matter how many queries are issued.

Sybase IQ has enhanced compression algorithms, which reduce the disk space necessary for data storing from 30 to 85 percent, depending on data's structure. A significant cost reduction results due to this fact. As already mentioned, storing data in a column-oriented manner increases the similarity of adjacent records on disk, and values within a column are often quite similar to each other. More tests confirmed that Sybase IQ requires 260 Terabytes of physical storage for storing 1000 Terabytes of row input data. A row-oriented database requires additional storage for indexes and, in the example above, this additional storage can reach up to 500 Terabytes [2]. In addition, Sybase IQ allows operating directly with compressed data [9], with a positive impact on processing speed.

As opposed to traditional databases, Sybase IQ is easier to maintain, and the tuning needed to get a higher performance requires less time and hardware resources. It doesn't need specialized schemas (dimensional modeling) to perform well. Sybase IQ is built upon open standards, so the integration and interoperability with other reporting systems and dashboards are easy to achieve. In order to enhance the performance of ad-hoc queries, it delivers more specialized indexes, such as: indexes for low cardinality data, grouped data, range data, joined columns, textual analysis, real-time comparisons for Web applications, date and time analysis [10]. Furthermore, every field of a row can be used as an index, without the need to define conventional indexes. Due to these indexes, analytical queries focus on specific columns, and only those columns are loaded into memory. This reduces very much the need for expensive, high performance disk storage and, at the same time, the number of I/O operations.

Offering an enhanced scalability, Sybase IQ can be used by a large number of users – hundreds and even thousands – which can access vast volumes of data, from a few gigabytes to several hundred terabytes.

Sybase IQ offers a fast and flexible access to information. As already mentioned, it is designed for query processing and ad-hoc analysis. As opposed to traditional data warehouses, it does not require data to be pre-aggregated in order to analyse it. Therefore, users can efficiently and quickly analyse atomic level data. With Sybase IQ users can analyse the business performance and can track the company's KPI (*key performance indicators*). Comparing with other products, it provides better solutions for measuring the business results, managing the customer relationship and ensuring financial controls.

Several intelligent features are integrated in Sybase IQ architecture. These features, such as the use of symmetric multiprocessing (SMP), enhance database's performance and reduce its maintenance overhead. A SMP architecture offers an increased performance because all processors can equally access the database's tables.

Sybase IQ architecture (figure 3) consists of multiple SMP nodes. Some of them are used only for reading data (*read-only nodes*), while other can be used both for reading and writing data (*read/write*

nodes). However, each node can be flexibly designated as a read or write node, according to the requirements at a given moment. Thus, if an overnight batch has to be executed in order to load or update a large volume of data, it's a good idea to make all the read/write nodes to operate as write nodes, even if they run as read nodes during the day. In addition, this architecture can be scaled up incrementally, by adding new nodes as needed.

As shown in figure 3, Sybase columnar store allows storing data in more repositories, depending on its age, and it can be easily scaled out. Data can be loaded through real time loads or batch ETL (*Extract, Transform, Load*) processes. Starting with the release of version 15 of Sybase IQ, a new "*load from client*" option has been added. This option allows loading data from external sources, via ODBC, JDBC, ADO.Net, OLE DB and DBLib. Data, which can be encrypted through different methods, is rapidly accessed for analytical or reporting purposes.

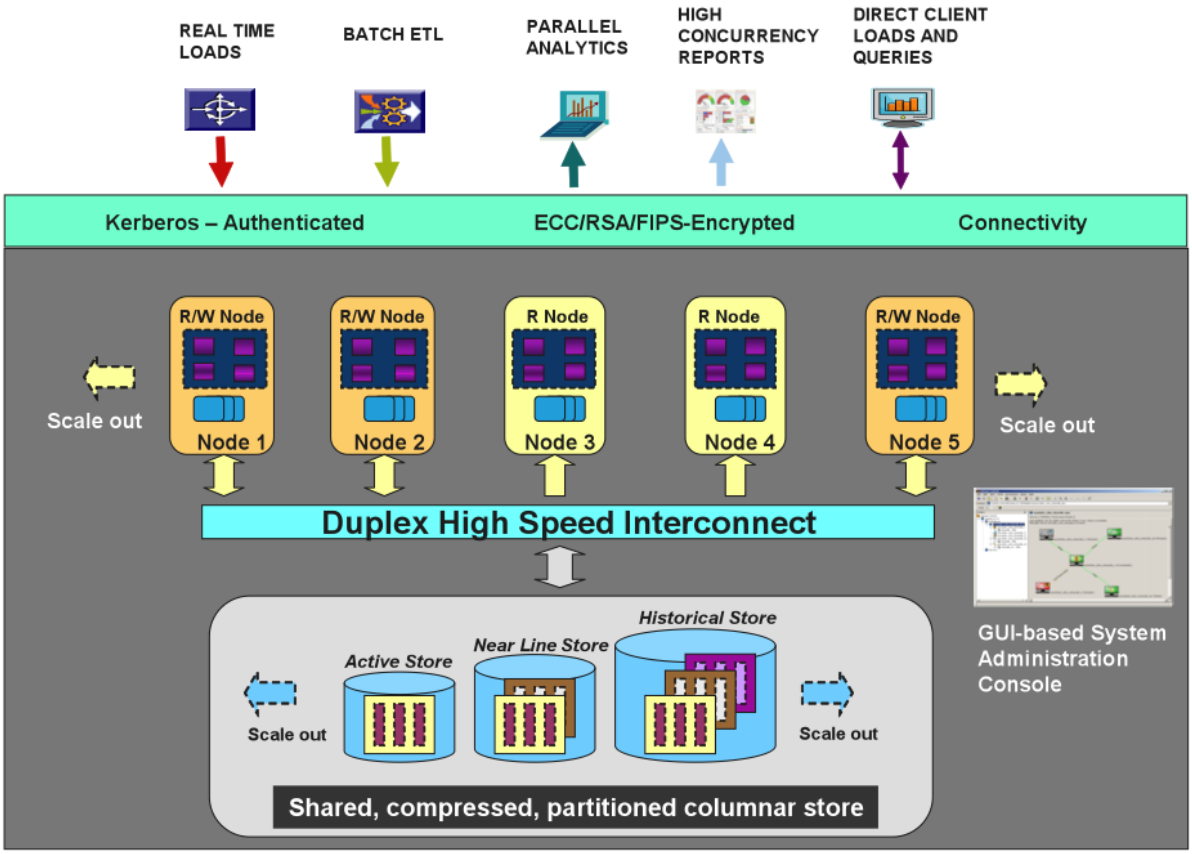

Fig. 3. Sybase IQ architecture
(source: [10])

Sybase IQ enables data to be managed more efficiently than in a traditional database, built in a row-oriented approach. Complex analyses are run much faster. High data compression reduces storage costs, and vast volumes of data can be processed much more quickly.

5.2.Vertica

To gain competitive advantages and comply with new regulations, companies are obliged to develop enterprise data warehouses and powerful applications able to respond to more and more ad-hoc queries from an increasing number of users that need to analyse larger volumes of data, often in real time.

Vertica Analytic Database is a DBMS that can help in meeting these needs. It is a column-oriented database that was built in order to combine both column store and execution, as opposed to other solutions that

are column-oriented only from storage point of view.

Designed by Michael Stonebraker, it incorporates a combination of architectural elements – many of them which have been used before in other contexts – to deliver a high-performance and low-cost data warehouse solution that is more than the sum of its elements.

Vertica is built in a massively parallel processing (MPP) architecture. In a MPP architecture, processors are connected with certain sets of data, data is distributed across the nodes of the network, and new processors can be added almost without limit. As data is partitioned and load into a server cluster, the data warehouse performs faster. Due to the MPP technology, the system performance and storage capacity can be enhanced simply by adding a new server to the cluster. Vertica automatically takes advantages of the new server without the need for expensive and time consuming upgrades.

While many of the new data warehouses use only MPP technology or columnar approach, Vertica is the only data warehouse that includes both innovations, as well as other features. It is designed to reduce I/O disk operations and is written natively to support grid computing.

Because of the columnar approach, which reduces the expensive I/O operations, queries are 50 to 200 times faster than a row-oriented database. Its MPP architecture offers a better scalability, that can be achieved by adding new servers in the grid architecture.

In order to minimize the disk space needed to store a database, Vertica uses more compression algorithms, depending on data type, cardinality and sort order. For each column, the proper algorithm is automatically chosen, based on a sample of the data [11]. As data into a column is homogenous, having the same type, Vertica provides a compression ratio from 8 to 13 time relative to the size of the original input data.

Vertica decomposes the logical tables and physically stores them as groups of columns named "*projections*". According to this concept, data is stored in different ways, similar to materialized views. Each projection contains a subset of the columns of one or more tables, and is sorted on a different attribute. Vertica automatically selects the proper projection in order to optimize query performance. Due to the effective compression algorithms used by Vertica, multiple projections can be maintained, which concurs to the performance improvement of a large range of queries, including ad-hoc queries needed for exploratory analysis.

On the other hand, these projections serve as redundant copies of the data. Because data is compressed so efficiently, Vertica can use the disk space to store these copies to ensure fault tolerance and to improve concurrent and ad-hoc query performance. Partitioning data across the cluster, Vertica ensures that each data element is stored on two or more nodes. Thus, an intelligent data mirroring is implemented, named K-Safety, where k is the number of node failures that a given set of projections will not affect the system availability. In order to guarantee K-Safety, k+1 replicas of all projections are built. Each replica has the same columns, but they may have different sort order. K-Safety allows requests for data stored into failed nodes to be satisfied by corresponding projections on other nodes. Once a failed node is restored, projections on the other nodes are automatically used to repopulate its data.

The value of k has to be configured so that a proper trade-off between hardware costs and availability guarantees to be met. If necessary, a new node can be added in the grid, and Vertica will automatically allocate a set of objects to that node and it can begin processing queries, increasing database performance [12]. Conversely, a node can be removed and the database will continue to work, but at a lower rate.

As shown in figure 4, a single Vertica node is organized into a *hybrid store* consisting of two distinct storage structures: the **Write-Optimized Store** (WOS) and the **Read-Optimized Store** (ROS). Generally, the WOS fits into main memory and is designed to efficiently support insert and update operations. Data is stored as collections of uncompressed and unsorted columns, maintained in update order.

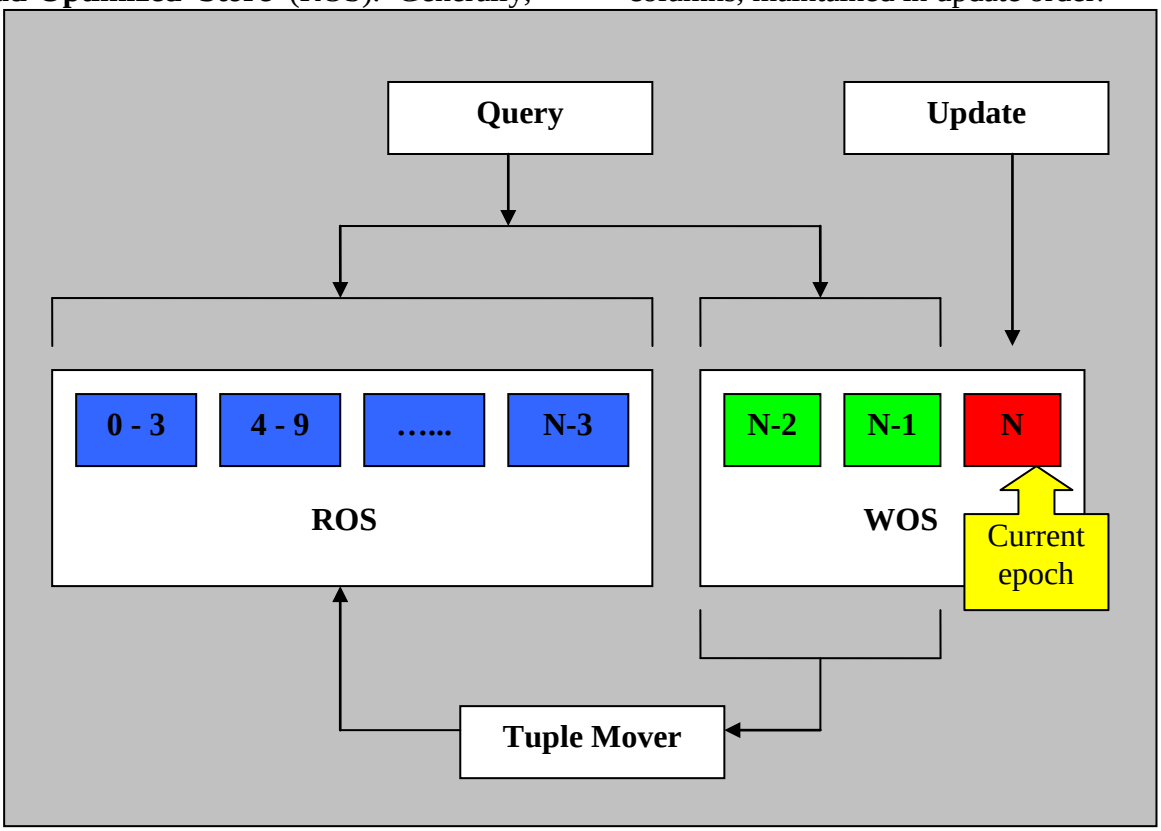

Fig. 4. Vertica storage model
(source: [11])

An asynchronous background process called the **Tuple Mover** moves data from WOS into the permanent disk storage in the ROS. The Tuple Mover operates on the entire WOS, sorting many records at a time and writing them to the ROS as a batch. Data in the ROS is sorted and compressed, so it can be efficiently read and queried.

Queries and updates do not interfere with one another. Updates are collected in time-based buckets called epochs. New updates are grouped in the current epoch until the transaction is committed. Data in older epochs is available for querying and moving into the ROS.

Because of the grid computing architecture, a query can by initiated on any node of the network. Vertica query planner decomposes the query according to the data stored into the involved nodes, and sends them the appropriate subqueries. Then it collects each node's partial result and composes them in order to offer to the requester the final answer.

In [13] a comparison is made between a 1.5 terabytes row-oriented data warehouse and a column-oriented database containing the same data and managed by Vertica Analytic Database. The results are presented in the table 1 bellow.

Table 1. Advantages of Vertica Analytic Database (source: [13])

	Row-oriented data warehouse	**Vertica Analytic Database**	**Vertica advantages**
Avg query response time	37 minutes	9 seconds	270x faster answers
Reports per day	30	1,000	33x more reports
Data availability	Next day	1 minute	Real-time views
Hardware cost	$1.4M (2*6 servers + SAN)	$50,000 (6 HP ProLiant servers)	$1/28^{th}$ of the hardware, built-in redundancy

All those presented above enable Vertica to manage larger volumes of historical data, analyse data at any level of detail, perform real-time analyses, conduct ad-hoc and short-lived business analytical projects, and build new analytic Software as a Service (SaaS) business.

6. Conclusions

For applications that write and update many data (OLTP systems), a row-oriented approach is a proper solution. In such an architecture, all the attributes of a record are placed contiguously in storage and are pushed out to disk through a single write operation. An OLTP system is a write-optimized one, having a high writing performance.

In contrast, an OLAP system, mainly based on ad-hoc queries performed against large volumes of data, has to be read-optimized. The repository of such a system is a data warehouse. Periodically (daily, weekly, or monthly, depending upon how current data must be), the data warehouse is load massively. Ad-hoc queries are then performed in order to analyse data and discover the right information for the decision making process. And for analytical applications, that read much more than they write, a column-oriented approach is a better solution.

Nowadays, data warehouses have to answer more and more ad-hoc queries, from a greater number of users which need to analyse quickly larger volumes of data.

Columnar database technology inverts the database's structure and stores each attribute separately, fact that eliminates the wasteful retrieval as queries are performed. On the other hand, much more data can be loaded in memory, and processing data into memory is much faster.

Column-oriented databases provide faster answers, because they read only the columns requested by users' queries, since row-oriented databases must read all rows and columns in a table. Data in a column-oriented database can be better compressed than those in a row-oriented database, because values in a column are much more homogenous than in a row. The compression of a column-oriented database may reduce its size up to 20 times, this thing providing a higher performance and reduced storage costs. Because of a greater compression rate, a column-oriented implementation stores more data into a block and therefore more data into a read operation. Since locating the right block to read and reading it are two of the most expensive computer operations, it's obviously that a column-oriented approach is the best solution for a data warehouse used by a Business Intelligence system developed for analytical purposes.

References

[1] C. Olofson, The Third Generation of Database Technology: Vendors and Products That Are Shaking Up the Market, 2010, www.idc.com

[2] Sybase, Sybase IQ: The Economics of Business Reporting, White paper, 2010, www.sybase.com/files/White_Papers/Sybase-IQ-Business-Reporting-051109-WP.pdf

[3] D. Loshin, Gaining the Performance Edge Using a Column-Oriented Database Management System, Sybase white paper, 2009, www.sybase.com

[4] Sybase, A Definitive Guide to Choosing a Column-based Architecture, White paper, 2008, www.information-management.com/white_papers/10002398-1.html

[5] W., McKnight, Evolution of Analytical Platforms, Information Management Magazine, May 2009, www.information-management.com/issues/2007_58/analytics_business_intelligence_bi-10015353-1.html

[6] D. Abadi, Column-Stores For Wide and Sparse Data, 3rd Biennial Conference on Innovative Data Systems Research, January 7 – 10, 2007, Asilomar, California, USA, http://db.csail.mit.edu/projects/cstore/abadicidr07.pdf

[7] D. Abadi, S. Madden, N. Hachem, Column-Stores vs. Row-Stores: How Different Are They Really?, Proceedings of the 2008, ACM SIGMOD International Conference on Management of Data, Vancouver, Canada, http://portal.acm.org

[8] M. Stonebraker, Daniel Abadi et al., C-Store: A column-oriented DBMS, Proceedings of the 31st VLDB Conference, Trondheim, Norway, 2005, http://db.csail.mit.edu/projects/cstore/vldb.pdf

[9] D. Tkach, When the information is the business, Sybase white paper, 2010, www.sybase.com/files/white_papers

[10] P. Howard, Sybase IQ 15.1, A Bloor InDetail Paper, 2009, www.it-director.com/business/innovation

[11] ***, Revolutionizing data warehousing in telecom with the Vertica Analytic Database, 2010, www.vertica.com/white-papers

[12] ***, The Vertica Analytic Database technical overview, 2010, www.vertica.com/white-papers

[13] ***, Increasing Enterprise Data Warehouse Performance, Longevity and ROI with the Vertica Analytic Database, 2010, www.vertica.com/white-papers

Gheorghe MATEI has graduated the Faculty of Planning and Economic Cybernetics in 1978. He achieved the PhD in Economic Cybernetics and Statistics in 2009, with a thesis on Business Intelligence systems in the banking industry. After a long career in the IT department, now he is working in the accounting and reporting department in Romanian Commercial Bank. His fields of interest include Business Intelligence systems, data warehousing, decision support systems, collaborative systems. He is a co-author of the book "*Business Intelligence Technology*" (2010), as well as author and co-author of several articles in journals, international databases and proceedings of national and international conferences in the mentioned domains.

Optimized Data Indexing Algorithms for OLAP Systems

Lucian BORNAZ
Faculty of Cybernetics, Statistics and Economic Informatics
Academy of Economic Studies, Bucharest
lucianbor@hotmail.com

The need to process and analyze large data volumes, as well as to convey the information contained therein to decision makers naturally led to the development of OLAP systems. Similarly to SGBDs, OLAP systems must ensure optimum access to the storage environment.
Although there are several ways to optimize database systems, implementing a correct data indexing solution is the most effective and less costly.
Thus, OLAP uses indexing algorithms for relational data and n-dimensional summarized data stored in cubes.
Today database systems implement derived indexing algorithms based on well-known Tree, Bitmap and Hash indexing algorithms. This is because no indexing algorithm provides the best performance for any particular situation (type, structure, data volume, application).
This paper presents a new n-dimensional cube indexing algorithm, derived from the well known B-Tree index, which indexes data stored in data warehouses taking in consideration their multi-dimensional nature and provides better performance in comparison to the already implemented Tree-like index types.
Keywords: *data warehouse; indexing algorithm; OLAP, n-Tree.*

1 Introduction

Data warehouses represented a natural solution towards increasing the availability of data and information, as well as their accessibility to decision makers. The warehouses store important data coming from different sources for later processing and are an integrant part of analytical processing systems (OLAP).
Unlike OLTP systems, OLAP systems must execute complex interrogations and large data volume analyses. To optimize, analytical processing systems analyze data and store aggregated information in special analytic structures, called cubes.
Similarly to OLTP systems, OLAP systems use indexing algorithms to optimize access to data stored in data warehouses, i.e. cubes.

2. General information about cubes

When stored in an OLAP system, the source data may be indexed to reduce the time necessary for their processing. To index source data, OLAP systems use indexing algorithms similar to OLTP (B-Tree, Bitmap, R-Tree etc.).
Processed data are stored in n-dimensional structures called cubes. The elements of a cube are the dimensions, members, cells, hierarchies and properties [1] (fig. 1).

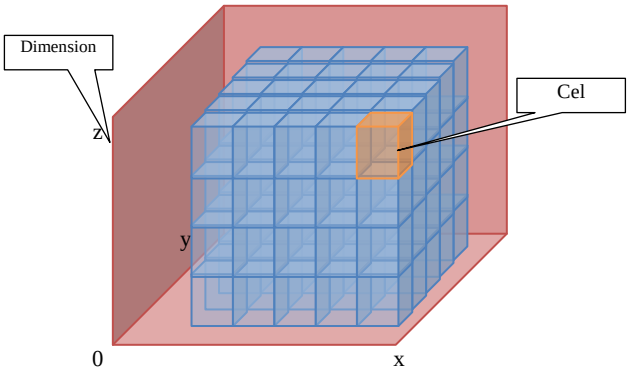

Fig. 1 - Structure of a tridimensional cube

The dimensions contain descriptive information about the data that is to be summarized. They are essential for data analysis and represent an axis of the cube [2].

Each dimension corresponds to a measure of the data source and uniquely contains each value stored in that position. During queries, the dimensions are used to reduce the search area and usually occur in the WHERE clause.

Hierarchies describe the hierarchical relationships between two or more members of the same dimension. A dimension can be part of multiple hierarchies. For example, in addition to the hierarchy of dimensions Quarter-Month-Year, Time dimension can belong to the hierarchy Day-Month-Year.

The cells of the cube contain summarized data based on dimensions values. Cells store summarized data based on the cube dimension number, dimensions values, method of analysis and is usually, the result returned by the queries.

Properties describe common features of all members of the same dimension. Properties allow selecting data based on similar characteristics. For example, the size of product volume may have an attribute which allows a certain volume products.

Analysis and data processing is based on the method chosen. The same data can be analyzed using different methods (clustering, neural nets, regression, Bayesian, Decision trees, etc.) accordingly to the user's needs. Although using different methods of analysis may result in different aggregate data and ordering different logic cells, the logical structure of the cube is the same.

In order to optimize performance, OLAP systems implement cube indexing algorithms. The indexes created through this process use the data contained in a cube's dimensions to quickly access the cells containing the data required by the user.

Hence, cubes are indexed using a B-Tree type of algorithm.

3. The B-Tree Index in OLAP Systems

As the values of the cube dimensions are unique and they are stored in the index blocks and used to locate the leaf blocks which contain references to the physical location of the cube cells, implementing a B-Tree index represents an effective solution for indexing cubes.

A B-Tree index used in OLAP systems contains sub-trees corresponding to each dimension.

The sub-trees are connected in such a way that each path to go through the tree from the root node to the final level index blocks (the ones storing the references to the cube's cells) is crossed by a sub-tree corresponding to each dimension.

Thus, a three dimensional cube contains three levels. The first level represents a matrix of planes (bi-dimensional space), the second level represents a matrix of lines (one dimensional space), and the third level represents a matrix of points in space (0 dimensional space) (fig. 2).

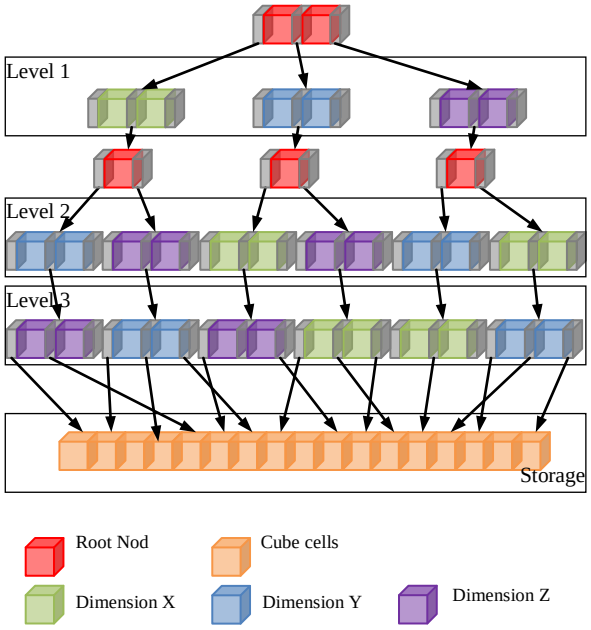

Fig. 2 - The structure of a B-Tree index for a three-dimensional cube [3]

Thus, each cube size will correspond to one sub-tree of the B-Tree index and each sub-tree will have on child sub-tree for each child dimension.

Summarized data are stored inside the cube, on pages separate from the cells. As summarized data creation and storage consumes CPU resources and storage space environment, the developer and/or system administrator can choose as they are available only at certain levels, depending on their usage, summarized data obtained from the upper levels being calculated by processing the data summarized in the lower levels.

Looking at the structure of the B-Tree index, it is easily noticeable that the cost of locating one of the cube's cells represents the sum of the costs associated with locating the last level index block of each sub-tree. The height of such an index is (f.1) [4]:

$$h \le \log_d \frac{r+1}{2} \qquad (f.1)$$

where $h \in N$, d is the number of stored values within an index block, r is the total number of values corresponding to the dimension.

The number of index blocks of a sub-tree containing d elements is:

$$\sum_{l=0}^{h} d^l = \frac{d^h - 1}{2d - 1} \qquad (f.2)$$

where $d=(m-1)$ and h is the tree height.

Considering that the number of items stored in index blocks at the top level of the sub-tree is r, we can calculate the maximum height (h) of the index portion corresponding to a specific dimension, as follows:

$$\frac{2d(d^h - 1)}{d - 1} \le r \quad => \quad 2d^h \le r + 1$$

$$=> \quad h \le \log_d \frac{r+1}{2} \qquad (f.3)$$

where $h \in N$.

The total cost of a search operation within a sub-tree is:

$$c_i = h+1 \qquad (f.4)$$

Since to locate a cell of the cube is necessary to cross every sub-tree, corresponding to each dimension of the cube, we can calculate the total cost of a queries based on (f.4):

$$c_{ti} = \sum_{j=1}^{n} c_{ij} + (d-1) \qquad (f.5)$$

where n is the number of cube dimensions, c_{ij} is the cost of query sub-tree corresponding to the dimension j and d-1 is the total number of root nodes used as connecting elements of the leaf blocks of sub-trees that have several subordinated sub-tree.

Since OLAP systems incorporate very large data volumes, their performance is affected not only by the query operations cost but also by the index storage space.

Indexes tend to occupy the storage space of the cube and sometimes their size can be larger than the data stored in the cube. If an index occupies a large memory space, it means that the structure is high (number of elements, elements that store too much data, etc.), which increases the index creation time and query execution times.

Using formulas (f.2) and (f.3) it can be calculated the storage space (SS) for a sub-tree indexed using a B-Tree index:

$$S_s = \frac{d^h - 1}{2d - 1} \cdot S_p \qquad (f.6)$$

where d is the number of items stored in a block index, h is the index height and SS is the page size.

Total storage space (St) needed to store a B-Tree index used for indexing cubes is equal to storage space for all its sub-trees. For a cube with three dimensions, the number of sub-trees of a B-Tree index is:

$$N_{si} = 1 + E_a + E_b + 2E_aE_b \qquad (f.7)$$

where N_{si} is the number of sub-trees of dimension i, E_a and E_b represents the elements number of dimension index leaf block corresponding to the other two dimensions.

Based on the (f.2)-(f.7), whole B-Tree index size can be calculated as:

$$S_t = \sum_{i=1}^{j} \left(S_{si} \cdot N_{si}\right) + S_{rn} \qquad (f.8)$$

where j is the cube dimensions number, S_{si} is a storage space needed to store the sub-tree for the dimension i, the N_{si} is the number of sub-trees coresponding to the dimension i and S_{rn} is the size of all nodes connecting the sub-trees.

The query cost of the B-Tree index is the sum of the cost of all sub-trees between the root node and the index leaf block which store the physical address of the cell.

4. The n-Tree Indexing Algorithm

Given the characteristics of cubes, as well as the structure of a B-Tree index, it becomes obvious that this indexing algorithm is not optimized for n-dimensional data structures. Thus, the number of sub-trees within the index is directly proportional with the number of dimensions. As a consequence, the cube is over-indexed resulting in an overconsumption of processing time and storage space.

The proposed n-dimensional indexing algorithm pays attention to the n-dimensional structure of the data. Instead of creating sub-trees corresponding to each dimension and subsequently linking them, it creates only one tree which indexes data simultaneously on all dimensions. As a result, the n-dimensional space is gradually divided into ever smaller n-dimensional subdivisions, until the smallest sub-divisions represent the cells of the cube.

The resulting index has the following characteristics:

- no NULL values are indexed;
- the root node contains at least two subordinated index blocks if it does not coincide with the last level index block;
- each index block contains:
- values from each dimension of the cube; the combination of such values represents a reference point in the n-dimensional space;

The index maintains an ordered list containing unique values corresponding to each dimension of the cube. The values in each list represent a subgroup of the values of the respective dimension. Combining values from each list at a time, we can obtain the data needed to identify the reference points in the n-dimensional space simultaneously minimizing the space required to store them.

Any value corresponding to a dimension from the subordinated index block is smaller than the value of the respective dimension corresponding to the reference point from the upper level index block.

- references to the subordinated index blocks (r_{bs}), corresponding to the reference points (f.9):

$$r_{bs} = \prod_{i=1}^{n} a_i \qquad (f.9)$$

where $a_{1..n}$ represents the number of values from dimensions 1..n stored in the index block;

- *n* references to the index blocks that contain larger values in a dimension than the reference point (one for each dimension).

Thus, an index block contains a total of r references (f.10):

$$r=r_{bs}+n \qquad (f.10)$$

where n equals the number of the cube's dimensions.

- the last level index blocks do not contain any references to other index blocks; instead they store the reference to the physical location of the cube's cells;
- the physical size of an index block is approximately one page.

Each index block stores an ordered list of unique values corresponding to each cube dimensions. Values from each list is a subset of these dimensions. Combining values from each list, one by one, points from the n-dimensional space can be identified, minimizing the needed storage space.

Any value from the dimensions, stored into an index block, is lower than the value belonging to the respective dimension from the reference point of the higher rank index block.

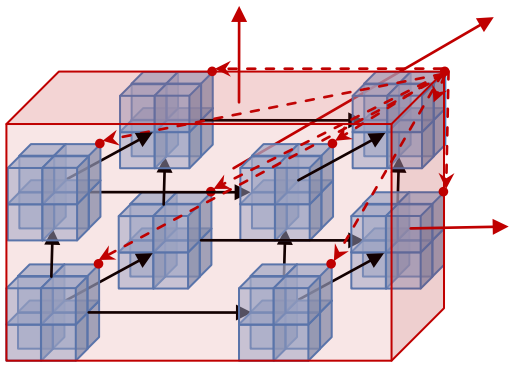

Fig. 3 - The structure of an n-dimensional index corresponding to a three dimensional cube

Because the dimensions values of the reference point are uniquely stored, the n-dimensional space is always a regular space. For a cube with three dimensions, this space is a rectangular parallelepiped, and ideally is a cube.

If some cells do not contain data units (containing null values), they are not indexed, thus reducing the size of the index. The lack of aggregated values corresponding to a cell does not affect the form of the space described by the values stored into a index block.

Every n-Tree index contains a sub-tree that indexes the summarized data which has the following characteristics:

- it contains a sub-tree for each dimension of the cube;

- each sub-tree corresponding to a dimension has a structure similar to a B-Tree index and indexes all the values pertaining to the respective dimension;
- the index blocks of the sub-trees contain references to the lower level index blocks;
- the index leaf blocks do not contain references to other indexing blocks; instead they are the sole elements containing references to the pages where the summarized data are stored;
- each element of the index leaf blocks contains references to parts of the n-dimensional space to which the respective value is assigned

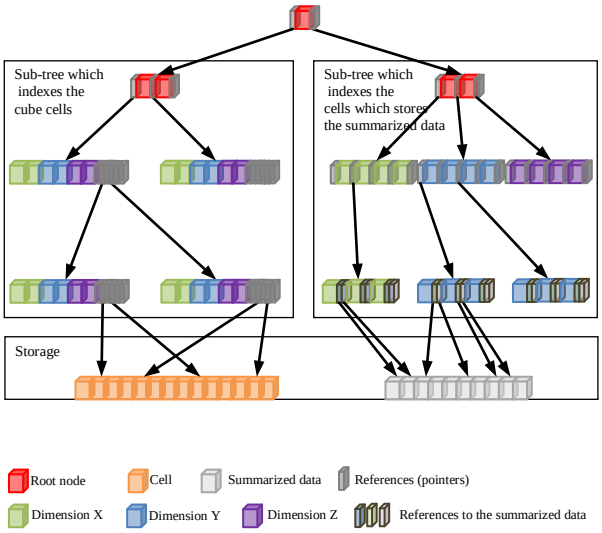

Fig. 4 - The structure of an n-Tree index for to a three dimensional cube

Thus the number of referenced contained by each element of a leaf index block is:

$$rs=n-1 \qquad (f.11)$$

where n is equal to the number of the cube's dimensions.

Summarized data relating to each value stored in the sub-tree corresponding to one dimension are equivalent to a 1 to (n-1) dimensional sub-space. The sub-spaces are distributed among the respective sub-trees, as to avoid storing multiple references to the same summarized data. The dimensions of the index are thus reduced.

For a 3-dimensional cube, the index leaf blocks will contain the following references:

- the elements of the index leaf block of the X dimension sub-tree contain:
- references to data summarized representing the space corresponding to the value of the X dimension, all values of the Y dimension and the first value of the Z dimension (one dimensional space);
- references to the data summarized representing the space corresponding to the value of the X dimension, all values of the Y dimensions and all values of the Z dimension (two dimensional space)
- the elements of the index leaf blocks of the sub-tree corresponding to the Y dimension contain:
- references to the data summarized representing the space corresponding to the value of the Y dimension, all values of the Z dimension and the first value of the X dimension (one dimensional space);
- references to the data summarized representing the space corresponding to the value of the Y dimension, all values of the X dimension and all the values of

the Z dimensions (two dimensional space);
- the elements of the index leaf blocks of the sub-tree corresponding to the Z dimension contain:
- references to the data summarized representing the space corresponding to the value of the Z dimension, all values of the X dimension and the first value of the Y dimension (one dimensional space);
- references to data summarized representing the space corresponding to the value of the Z dimension, all values of the X dimensions and all values of the Y dimension (two dimensional space).

5. Creating an n-Tree Index

To create an n-dimensional index, all data in every index is read and n-dimensional points are created. For each of these points, the following operations are carried out:

- an index block corresponding to an n-dimensional sub-space whose reference point has only values larger than that of the processed point is identified; the index block must also have enough free space to store the values of the corresponding dimensions of the processed point plus a reference;

If such an index block is identified, the values are added to the dimensions' corresponding lists and the reference to the physical location of the cube's cell is stored. Otherwise, a new index block is created by dividing one of the neighboring index blocks.

- when a new index block is created, the values of the reference point, as well as the reference to the parent index block are added, together with references to the neighboring index blocks;

$$S_e = \sum_{i=1}^{j} S_{vi} + S_{ref} \qquad (f.12)$$

where S_e is the element size, S_{vi} is the data type size for the dimension i and S_{ref} is the size of a reference.

This process could propagate itself to the root node. Generally, OLAP systems contain historical data with a low frequency of updating operations but with a large volume of updates. Updating these data also triggers an updating of the cube, and thus, of its index.

It should be noted that the space required for inserting a new element into a block index is not always the same. If some values of n-dimensional point corresponding to a specific dimension were previously inserted the necessary free space is smaller than the element size.

6. The Performance of the n-Tree Index

The performance of an index depends on its height. A larger height means more physical read operations are needed to identify the cell containing the required data.

The height of the index depends on the size of the index block, the size of the type of indexed data, the number of references to subordinated index blocks stored in each index block, and the number of cells.

By analyzing the structure of an index block (fig. 4), we can compute the number of references it can store.

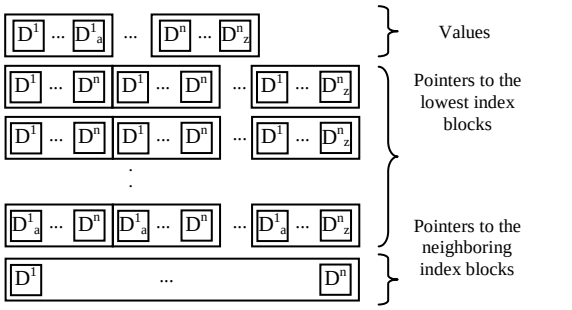

Fig. 5 - Structure of an index block in an n-Tree index

The volume of the stored data in an index block may be written as (f.11):

$$S_d = \sum_{i=1}^{n} a_i \cdot S_v + \left(\prod_{i=1}^{n} a_i + n \right) \cdot S_{ref} \qquad (f.13)$$

where S_d represents the volume of the stored data, S_v represents the size of the indexed value, n is the number of dimensions and S_{ref} the size of a reference.

The blocks number of the n-Tree index which contains d elements is:

$$\sum_{l=0}^{h} d^l = \frac{d^h - 1}{2d - 1} \qquad (f.14)$$

where h is the index height.

Ideally, d is the maximum number of the elements which can be stored inside an n-Tree index and its value is up to $\prod_{i=1}^{n} a_i + n$.

The query cost for the n-Tree index is:

$$c_i = h + 1 \qquad (f.15)$$

where c_i is the query cost, h is the index height and 1 is for the index root block.

Since the data in an OLAP system is rarely modified, the best performance is obtained when the index blocks contain a volume of data equal to their size.
Therefore, we can approximate the value of S_d to be equal to that of a page.
We assume that:

- the size of an index block is of 8kB (this is the most common size in current database systems [5]);
- the size of a reference is 6B (the most common size for a local index [6]);
- the size of the data type is 8B (this is the size of the *datetime* type of data);
- each dimension contains the same number of unique values.

Using the formulas (f.3) and (f.15), we can compare the performance of a n-Tree index to that of a B-Tree index (fig 6-11).

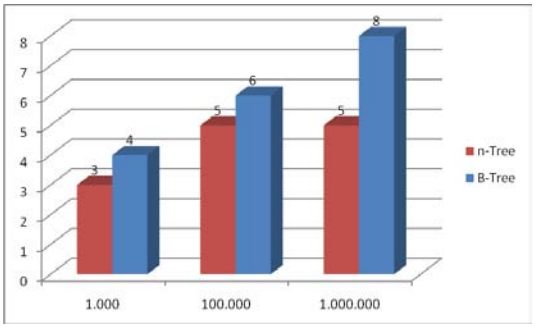

Fig. 6 - The cost of a search operation in a two-dimensional cube

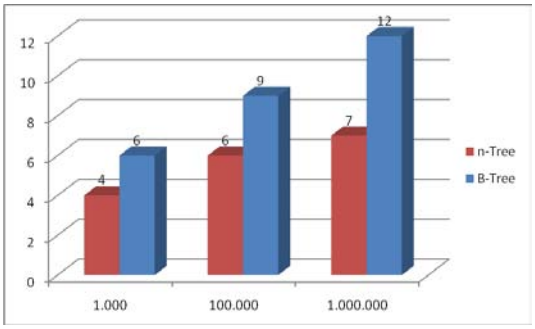

Fig. 7 - The cost of a search operation in a three dimensional cube

Analyzing fig. 6 and fig 7, it can be seen that the n-Tree index query cost is lower the the B-Tree index query cost event for 1.000 cells. The performance difference is event higher when the cells number or the dimensions number increase.

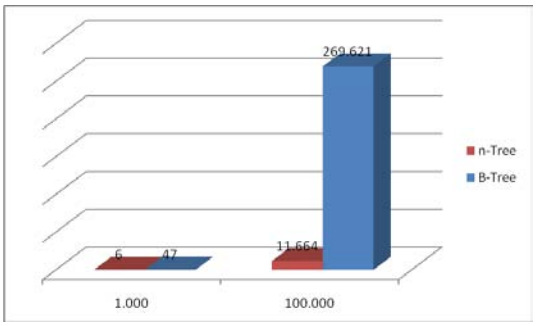

Fig. 8 - The size [in MB] of an index corresponding to a two dimensional cube

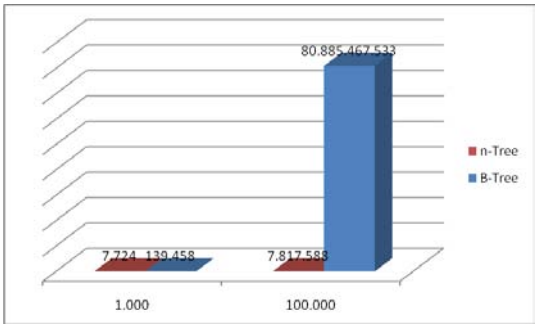

Fig. 9 - The size [in MB] of an index corresponding to a three dimensional cube

The same situation can be observed for the index size in fig. 8 and fig. 9.
The n-Tree index is much smaller than the B-Tree index. The difference comes from the lower number of the index blocks and from the flexibility of creating the summarized data.
When the data summarized data is to be query, the result depends on the location of the location of the summarized data.

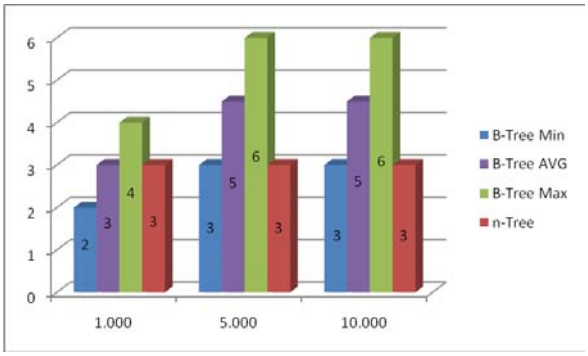

Fig. 10 - The summarized data query cost for a two-dimensional cube

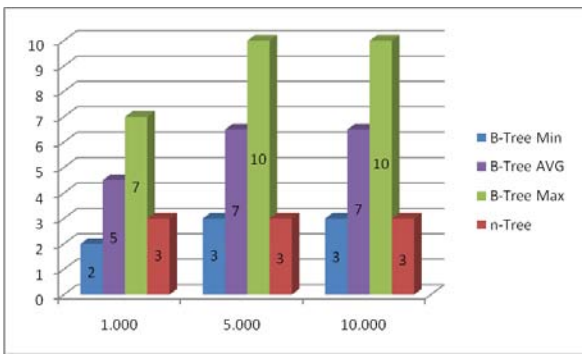

Fig. 11 - The summarized data query cost for a three-dimensional cube

Anyway, in fig. 10 and fig. 11 it can be observed that the n-Tree index query cost is lower than the B-Tree index query cost in any situation, excepting when the cells number is very low.

7. Conclusions

Implementing a new indexing algorithm, with much wider scope and increased flexibility, could be the database systems optimization solution, especially when other indexing algorithms do not provide the desired results.
The n-Tree index could be considered a more generalized B-tree index. If B-Tree index can index only uni-dimesional data, the n-Tree index is optimized for any n-dimensional data. Moreover, the n-Tree index will be more suitable for indexing spatial data.
As shown in figures 5-9, the n-Tree index outperforms the B-Tree index in locating the cells of the cube. Moreover, the difference in performance increases as the number of the cube's cells rises. In addition, the space occupied by the n-Tree index in much smaller than that needed for a B-Tree index. Again the superiority of the n-Tree index is all the more evident when the number of the cube's cells increases.

References

[1] Revista Informatica Economica, nr. 4 (24), 2002;
[2] Revista Informatica Economica, nr. 1 (17), 2001;
[3] Computing Partial Data Cubes for Parallel Data Warehousing Applications, Frank Dehne, Andrew Rau-chaplin, Computational Science - ICCS 2001;
[4] „Ubiquitous B-Tree", Douglas Corner, ACM, 1979;
[5] MCTS 70-431: Implementing and Maintaining Microsoft SQL Server 2005, Que, 2006;
[6] „Index Internals", Julian Dyke, 2005.

Lucian BORNAZ is PhD candidate at Academy of Economic Studies since 2006.
His research domain is database systems with focus on data indexing algorithms.
He graduated Airforce Academy in 1998 and master course at Academy of Economic Studies in 2006.
Lucian Bornaz is certified by Microsoft as MCP, MCSA, MCSE, MCDBA, MCAD and MCTS.

Cost Effective RADIUS Authentication for Wireless Clients

Alexandru ENACEANU
Faculty of Computer Science, Romanian-American University, Bucharest, Romania
alexandru.enaceanu@profesor.rau.ro
Gabriel GARAIS
Faculty of Computer Science,
Romanian-American University, Bucharest, Romania

Network administrators need to keep administrative user information for each network device, but network devices usually support only limited functions for user management.
WLAN security is a modern problem that needs to be solved and it requires a lot of overhead especially when applied to corporate wireless networks. Administrators can set up a RADIUS server that uses an external database server to handle authentication, authorization, and accounting for network security issues.
Keywords: *RADIUS, WLAN, Wireless Authentication, Wireless Access Control*

1 Introduction Corporate wireless networks are in general the primary source of hacking into the corporate systems. The risks to users of wireless technology have increased lately, as the service has become more popular. There were relatively few dangers when wireless technology was first introduced. Crackers had not yet had time to latch on to the new technology and wireless was not commonly found in the work place. However, there are a great number of security risks associated with the current wireless protocols and encryption methods.

Many early access points cannot discern whether or not a particular user has authorization to access the network. Although this problem reflects issues that have long troubled many types of wired networks (it has been possible in the past for individuals to plug computers into randomly available Ethernet outlets and get access to a local network), this did not usually pose a significant problem, since many organizations had reasonably good physical security. However, the fact that radio signals travel outside of buildings makes physical security largely irrelevant to hackers.

Common wireless encryption using WEP or a WPA key combined with static MAC entries is good for small offices, but totally inadequate for a corporate wireless or a campus network. That is because a corporate wireless network has a lot of access points to reconfigure for changing the access key or adding a new MAC address to the allowed clients list.

WLAN security can be significantly strengthened by using 802.1X to control access point access and deliver dynamic keys to authenticated users. Authentication Servers based on the RADIUS protocol play a key role in 802.1X [1].

2. Authentication, Authorization, and Accounting

2.1 RADIUS Protocol

The Remote Authentication Dial In User Service (RADIUS) protocol was originally defined to enable centralized authentication, authorization, and access control (AAA) for SLIP and PPP dial-up sessions. Instead of requiring every Network Access Server (NAS) to maintain a list of authorized usernames and passwords, RADIUS Access-Requests were forwarded to an Authentication Server, commonly referred to as an AAA Server (AAA stands for authentication, authorization, and accounting). This architecture made it possible to create a central user database, consolidating decision-making at a single

point, while allowing calls to be supported by a large, physically distributed set of NASs.[2]

When a user connects, the NAS sends a RADIUS Access-Request message to the AAA Server, relaying information like the user's name and password, type of connection (port), NAS identity, and a message Authenticator [3].

Upon receipt, the AAA Server uses the packet source, NAS identity, and Authenticator to determine whether the NAS is permitted to send requests. If so, the AAA Server tries to find the user's name in its database. It then applies the password and perhaps other attributes carried in the Access-Request to decide whether access should be granted to this user. Depending upon the authentication method being used, the AAA Server may return a RADIUS Access-Challenge message that carries a random number. The NAS relays the challenge to the remote user (for example, using CHAP). The user must respond with the correct value to prove its asserted identity (for example, encrypting the challenge with its password), which the NAS relays to the AAA Server inside another RADIUS Access-Request message.[2]

If the AAA Server verifies that the user is authentic and authorized to use the requested service, it returns a RADIUS Access-Accept message. If not, the AAA Server returns a RADIUS Access-Reject message and the NAS disconnects the user. [3]

2.2 RADIUS and Wireless LANs

In a wireless network that uses 802.1X Access Control, the wireless station plays the role of the Remote User and the wireless access point plays the role of the Network Access Server. Instead of connecting to the NAS with a dial-up protocol like PPP, wireless stations associate to the access point using 802.11 protocols.

If the AAA Server issues an Access-Accept message, the access point and wireless station complete a handshake to generate session keys used by WEP or TKIP to encrypt data. At that point, the access point unblocks the port and the wireless station can send data and receive data to and from the attached network. If the AAA Server issues an Access-Reject message, the access point disassociates the station. The failed station can try to authenticate again, but the access point prevents the station from actually sending data through the access point into the adjacent network.

2.3 Cost of RADIUS servers

There are many options on the market for setting up a RADIUS server.

RADIUS server software has a price range from $400 and up to several thousand dollars, depending on your implementation, number of clients and reports.

Commercial RADIUS Servers for a campus network or mid-sized organization vary in price as following:

- Interlink RADIUS server cost about $2375; [7]
- $2800 is also the cost for one Odyssey Server; [8]
- VOP Radius Small Business starts at $2100; [9]
- Radiator license will cost about $720. [10]

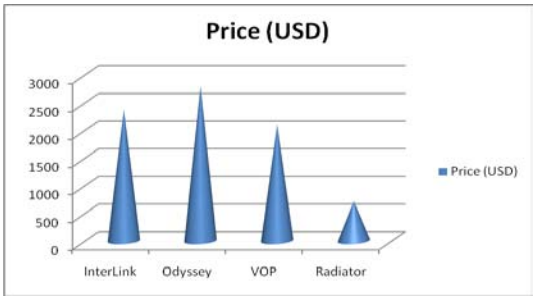

Fig. 1. Price comparison for commercial RADIUS software

Taking into consideration the above mentioned prices and also the specifications of each software package, Radiator is the most cost-effective solution for a campus network.

RADIUS servers are also available in hardware/software combo packages.

For example, a Juniper Networks Steel-Belted Radius (SBS) solution for a campus network is available for $7500. [8]
Meanwhile LeapPoint AiroPoint 3600-SE appliance starts at $12000. [9]

Fig. 2. Price comparison for commercial RADIUS combo packages

The above high priced combo solutions are least cost-effective solutions when compared to open source applications. A very cost effective RADIUS solution is FreeRADIUS. FreeRADIUS is a powerful RADIUS server on Linux from the open source community which can fit in today's distributed and heterogeneous computing environment. FreeRADIUS supports LDAP, MySQL, PostgreSQL, and Oracle databases and is compatible with such network protocols as EAP and Cisco LEAP. FreeRADIUS is currently being deployed in many large-scale production network systems being very efficient.

2.4 Installing and configuring FREERADIUS

Depending on your Linux distribution you can download FreeRADIUS as a binary package or as a source. Both methods are straightforward and require minimum Linux operating system knowledge.
Building from source implies the following steps:

- Download the latest archive from ftp://ftp.freeradius.org
- Unzip and untar the archive
- Run the following commands: *./configure ; make; make install*

Configuring the RADIUS server consists of configuring the server, the client, and the user (both for authentication and authorization). There can be different configurations of the RADIUS server for different needs; fortunately most of the configurations are similar.

- **Configuring the server**

FreeRADIUS configuration files are usually stored in the /etc/raddb folder. First we need to modify the radiusd.conf file and uncomment the *$INCLUDE sql.conf* line. Other options from *radiusd.conf* file should then look like:

```
authorise {
          preprocess
          chap
          mschap
          suffix
          eap
          files
          sql
          pap
   }

accounting {
          detail
          sql
   }
```

Next, we have to edit /etc/raddb/sql.conf, and direct it to the appropriate database (PostgreSQL, MySQL, etc.), by modifying the line: database = "mysql".
Also we need to edit the connection zone and specify the IP address of the SQL server we are going to use, the port and login credentials:

```
server = "a.b.c.d"
port = 3306
login = "raduser"
password = "radpass"
```

Database table name configuration for PostgreSQL, MySQL can be altered using the following line *radius_db = "radius"*.
If using Oracle, then the above line changes into:

```
radius_db=
"(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HO
ST=localhost)(PORT=1521))(CONNECT_DATA=(
SID=your_sid)))"
```

[6]

Clients are configured in */etc/raddb/clients.conf*. There are two ways to configure RADIUS clients. You can group the NAS by IP subnet or you can list the NAS by hostname or IP address [4]:

- Grouping the NAS by IP subnet

```
client 192.168.0.0/24 {
   secret          = mysecret1     - the
"secret"   should   be   the   same   as
configured on NAS
   shortname     = mylan           - the
"shortname" can be used for logging
   nastype         = cisco         - the
"nastype" is used for checkrad and is
optional
}
```

- Listing the NAS by hostname or IP address

```
client 192.168.0.1 {
   secret      = mysecret1
   shortname   = myserver
   nastype      = other
}
```

- ***Setting up RADIUS database***

First, we should create a new empty 'radius' database in SQL and a database user with permissions to that database. The user should be the same as specified above in the login credentials.

Next, we need to create the schema for your database. There is an SQL script file for each SQL type in doc/examples/ in operating system's doc directory (or where FreeRADIUS has been unzipped) [5].

Create MySQL Database

```
mysql –u root -p
  CREATE DATABASE radius;
  GRANT     ALL     ON     radius.*     TO
radius@localhost      IDENTIFIED      BY
"radpass";
  exit
mysql -uroot -p radius < mysql.sql
```

Create PostgreSQL Database

```
su - postgres
createuser radius --no-superuser --no-
createdb --no-createrole -P
createdb radius --owner=radius
exit
psql -U radius radius < postgresql.sql
```

- ***Populating SQL database***

Creating RADIUS users is straightforward, as data need to be inserted in the below tables accordingly:

- *In usergroup*, put entries matching a user account name to a group name.
- In *radcheck*, put an entry for each user account name with a 'Cleartext-Password' attribute with a value of their password.
- In *radreply*, create entries for each user-specific radius reply attribute against their username
- In *radgroupreply*, create attributes to be returned to all group members

At a minimum, only for authentication purpose with no options, the only table we need to edit is *radcheck.*

A simple RADIUS authentication record will look like:

id	username	attribute	op	value
1	Alex	Cleartext-Password	:=	password123

Fig. 3. Simple RADIUS SQL authentication record

For ease-of-use we can build a script to add new users or we can use another open-source product, daloRADIUS.

daloRADIUS is an advanced RADIUS web management application aimed at managing hotspots and general-purpose ISP deployments. It features user management, graphical reporting, accounting, a billing engine and integrates with GoogleMaps for geo-locating.

Installing and configuring daloRADIUS is beyond the scope of this article.

- **Test and implement**

Test the configuration by reloading FreeRADIUS and then run the following command:

```
radtest user passwd radius-server[:port]
nas-port-number secret.
```

Example:

```
radtest Alex password123 localhost 1
testing123
```

will produce the following result:

```
Sending Access-Request of id 17 to
127.0.0.1 port 1812
        User-Name = "Alex"
        User-Password = "password123"
        NAS-IP-Address = 127.0.1.1
        NAS-Port = 1
```

Configuring the access points to use the RADIUS server for accouting and access control is straightforward and the required fields would be: *server IP, server port, shared secret.*

2.5 FreeRADIUS performance test

The database containing Freeradius records is very simple, contains neither transactions, nor triggers, nor views. Therefore, the criteria used for the performance test is disponibility and speed will be key factors to take a decision.

MySQL and PostgreSQL both offer some solutions – similar – to answer to needs of applications requiring high disponibility. Replication seems to be a good choice as it is possible to specify different servers for accounting and authentication (write and read). Write accesses can be sent to the Master and reads to the slaves.

As both databases offer high availability options, the choice is made based on the queries speed. MySQL is popular for its speed (especially for reads) but PostgreSQL's robustness in a concurrential environment would be better.

We determined the number of transactions that each database is able to perform in a given time. Results are compared to a basic setup that gets usernames from a text file. Conditions will be strictly identical:

- We used the same machine and the same operating system;
- Tables structure contains identical indexes;
- Pools of 50 connections are created.

We measure the execution time to authenticate 10000 users with the *radclient* tool in concurrent access.

time /usr/local/bin/radclient -p 1000 -q -s -f radius.test 127.0.0.1 Alex password123

Results are grouped as following:

Database	File	Postgre SQL	My SQL
Time (sec)	76	25	9
Transactions /second	131	400	1111

Reading from PostgreSQL as well as MySQL, is much faster than in a text file. This can be explained by indexes creation within tables. This difference would be smaller for a database containing a much smaller number of users. MySQL realises excellent results processing three times more transactions than PostgreSQL. Results would be close to these in a Master-Slave environment where reads would be sent to another server than writes.

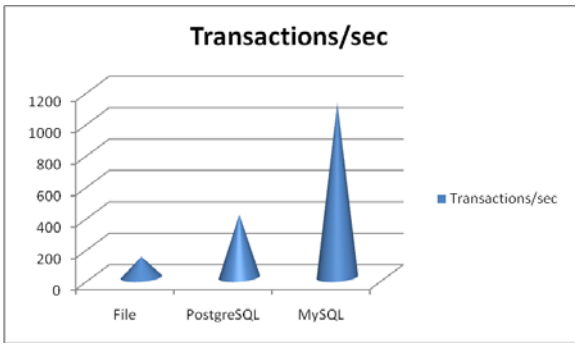

Fig. 4. DB performance chart

Conclusion

By following the steps outlined in this article, administrators can set up a RADIUS server that uses an external SQL server to handle authentication, authorization, and accounting for network security issues, by using a very cost-effective approach.

This article has provided the following:

- An introduction to the RADIUS and SQL servers and to the AAA concept;
- A scenario to put the installation and implementation into context;
- Options and cost for various RADIUS servers;
- Instructions on installing and configuring the RADIUS server;
- Details on configuring the network access server;
- A sample of the detailed information that RADIUS will provide and manage;
- DB performance test.

By following directions from this article, network administrators ease their task of making sure protected data can only be accessed by authorized entities on wireless networks.

References

[1] Jonathan Hassell, Securing Public Access to Private Resources, O'Reilly Media, 2002
[2] RADIUS described by RFC 2865 of the Internet Engineering Task Force http://www.ietf.org/rfc/rfc2865.txt
[3] Cisco, RADIUS configuration http://www.cisco.com/en/US/docs/ios/12_1/security/configuration/guide/scdrad.html
[4] Building a RADIUS server on Linux , IBM Software group, May 2005 http://www.ibm.com/developerworks/library/l-radius/
[5] FreeRADIUS Homepage http://www.freeradius.org
[6] FreeRADIUS with Oracle. http://www.ioncannon.net/system-administration/136/freeradius-with-oracle/
[7] Interlink Networks, http://www.interlinknetworks.com/
[8] Juniper Networks, http://www.juniper.net
[9] ServerWatch, http://www.serverwatch.com
[10] Radiator Server, http://www.open.com.au/radiator/

Alexandru ENACEANU is assistant lecturer at Romanian American University, faculty of Managerial Informatics since year 2003 and Ph.D. student of the Faculty of Cybernetics, Statistics and Economic Informatics. Amongst his fields of interest and didactic activities are networking, security, web technologies and database management systems.

Gabriel Eugen GARAIS is assistant lecturer at Romanian American University, faculty of Managerial Informatics since year 2002 and Ph.D. student of the Faculty of Cybernetics, Statistics and Economic Informatics. Amongst his fields of interest and didactic activities are web technologies and database management systems.

Database Replication

Marius Cristian MAZILU
Academy of Economic Studies, Bucharest, Romania
mariuscristian.mazilu@gmail.com, mazilix@yahoo.com

For someone who has worked in an environment in which the same database is used for data entry and reporting, or perhaps managed a single database server that was utilized by too many users, the advantages brought by data replication are clear. The main purpose of this paper is to emphasize those advantages as well as presenting the different types of Database Replication and the cases in which their use is recommended.
Keywords: *Database Replication, Snapshot replication, Merge replication, Transactional replication*

1 Introduction

Imagine a scenario in which you have to develop an application that all the company's staff will use to perform different tasks. Each person has a laptop and will be connected to the company's network.

This type of application can be developed in two different ways.

One of those is the traditional approach of separating the tables from the other objects in the database so that the data can reside in a back-end database on a network server, or on the Internet or an intranet, while the queries, forms, reports, macros, and modules reside in a separate front-end database on the user's computer. The objects in the front-end database are based on tables that are linked to the back-end database. When users will retrieve or update information in the database, they use the front-end database.

The second way enables you to take a new approach to building this solution by creating a single database that contains both the data and objects. Using Database replication, you can then make replicas of the database for each user and synchronize each replica with the Design Master on a network server.

In this scenario, you can choose to replicate only a portion of the data in the Design Master, and you can replicate different portions for different users by creating *partial replicas*. By using partial replicas, you can duplicate only the data that each user actually needs. A complete set of data is still contained in the Design Master, but each replica handles only a subset of that data.

The Design Master is the first member in a replica set and it is used in the creation of the first replica in a replica set. You can make changes to the database structure only with the Design Master.

Replicas in the same replica set can take turns being the Design Master, but there can be only one Design Master at a time in each replica set.[1]

2 The concept of Replication

To better understand the method behind Database Replication we start with the term "Replication" which represents the process of sharing information to ensure consistency between redundant resources, such as software or hardware components, to improve reliability, fault-tolerance, or accessibility. It could be *data replication* if the same data is stored on multiple storage devices, or *computation replication* if the same computing task is executed many times.

The access to a replicated entity is typically uniform with access to a single, non-replicated entity. The replication itself should be transparent to an external user. In addition, in a failure scenario, a failover of replicas is hidden as much as possible.

In systems that replicate data the replication itself is either *active or passive.*

We talk about an *active replication* when the same request is processed at every replicated instance and about *passive replication* when each request is processed on a single replica and then its state is transferred to the other replicas.

If at any time one master replica is designated to process all the requests, then we are talking about the *primary-backup* scheme (*master-slave* scheme) predominant in high-availability clusters.

On the other side, if any replica processes a request and then distributes a new state, then this is a *multi-primary* scheme (called *multi-master* in the database field). [2]

Even thought the process of Data Replication it's used to create instances of the same or parts of the same data, we must not confuse it with the process of backup since replicas are frequently updated and quickly lose any historical state. Backup on the other hand saves a copy of data unchanged for a long period of time.

3 What is Database Replication

Database replication is the process of creating and maintaining multiple instances of the same database and the process of sharing data or database design changes between databases in different locations without having to copy the entire database.

In most implementations of database replication, one database server maintains the master copy of the database and the additional database servers maintain slave copies of the database. The two or more copies of a single database remain synchronized. [3]

The original database is called a *Design Master* and each copy of the database is called a *replica*. Together, the Design Master and the replicas make up a *replica set*. There is only one Design Master in a replica set.

Synchronization is the process of ensuring that every copy of the database contains the same objects and data. When you synchronize the replicas in a replica set, only the data that has changed is updated. You can also synchronize changes made to the design of the objects in the Design Master. [1]

Database writes are sent to the master database server and are then replicated by the slave database servers.

Database reads are divided among all of the database servers, which results in a large performance advantage due to load sharing.

In addition, database replication can also improve availability because the slave database servers can be configured to take over the master role if the master database server becomes unavailable. [3]

4 When to chose Database Replication

Implementing and maintaining replication might not be a simple proposition. If you have numerous database servers that need to be involved in various types of replication, a simple task can quickly become complex.

Implementing replication can also be complicated by the application architecture. However, there are numerous scenarios in which replication can be utilized.[4]

Database replication is well suited to business solutions that need to:

- ***Share data among remote offices.*** You can use database replication to create copies of a corporate database to send to each satellite office across a wide area network (WAN). Each location enters data in its replica, and all remote replicas are synchronized with the replica at corporate headquarters. Individual replicas can maintain local tables that contain information not included in the other replicas in the set.
- ***Share data among dispersed users.*** New information that is entered in the database while users are out of the office can be synchronized any time the users establish an electronic link with the corporate network. As part of their workday routine, users can dial in to the network, synchronize the replica, and work on the most current version of the database. Because only the incremental changes are transmitted during synchronization, the time

and expense of keeping up-to-date information are minimized. By using partial replicas, you can synchronize only specified parts of the data.

- ***Make server data more accessible.*** If your solution does not need to have immediate updates to data, you can use database replication to reduce the network load on your primary server. Introducing a second server with its own copy of the database improves response time. You determine the schedule for synchronizing the replicas, and you can adjust that schedule to meet the changing needs of your users. Replication requires less centralized administration of the database while offering greater access to centralized data.
- ***Distribute solution updates.*** When you replicate your solution, you automatically replicate not only the data in your tables, but also your solution's objects. If you make changes to the design of the database, the changes are transmitted during the next synchronization; you don't have to distribute complete new versions of the software.
- ***Back up data.*** At first glance, database replication might appear to be very similar to copying a database. However, while replication initially makes a complete copy of the database, thereafter it simply synchronizes that replica's objects with the original objects at regular intervals. This copy can be used to recover data if the original database is destroyed. Furthermore, users at any replica can continue to access the database during the entire backup process.
- ***Provide Internet or intranet replication.*** You can configure an Internet or intranet server to be used as a hub for propagating changes to participating replicas.[1]

5 When Database Replication should not be used

Although database replication has many benefits and can solve many problems in distributed-database processing, we should recognize the fact that in some situations replication is less then ideal. Database Replication is not recommended if:

- ***There are frequent updates of existing records at multiple replicas.*** Solutions that have a large number of record updates in different replicas are likely to have more record conflicts than solutions that simply insert new records in a database. If changes are made to the same record by different users and at the same time then record conflicts will definitely appear. This can be real time consuming because the conflicts must be resolved manually.
- ***Data consistency is critical at all times.*** Solutions that rely on information being correct at all times, such as funds transfers, airline reservations, and the tracking of package shipments, usually use a transaction method. Although transactions can be processed within a replica, there is no support for processing transactions across replicas. The information exchanged between replicas during synchronization is the result of the transaction, not the transaction itself.

6 Methods of performing Database Replication

Database replication can be performed in at least three different ways:

- ***Snapshot replication****:* Data on one database server is plainly copied to another database server, or to another database on the same server.
- ***Merging replication****:* Data from two or more databases is combined into a single database.
- ***Transactional replication****:* Users obtain complete initial copies of the database and then obtain periodic updates as data changes.

6.1 Snapshot replication

This type of Database Replication is one of the simplest method to set up, and perhaps the easiest to understand.

The snapshot replication method functions by periodically sending data in bulk format. Usually it is used when the

subscribing servers can function in read-only environment, and also when the subscribing server can function for some time without updated data. Functioning without updated data for a period of time is referred to as *latency*.

For example, a retail store uses replication as a means of maintaining an accurate inventory throughout the district. Since the inventory can be managed on a weekly or even monthly basis, the retail stores can function without updating the central server for days at a time. This scenario has a high degree of latency and is a perfect candidate for snapshot replication.

Additional reasons to use this type of replication include scenarios with low-bandwidth connections. Since the subscriber can last for a while without an update, this provides a solution that is lower in cost than other methods while still handling the requirements.

Snapshot replication also has the added benefit of being the only replication type in which the replicated tables are not required to have a primary key.

Snapshot replication works by reading the published database and creating files in the working folder on the distributor. These files are called snapshot files and contain the data from the published database as well as some additional information that will help create the initial copy on the subscription server.[5]

Snapshot replication is often used when needing to browse data such as price lists, online catalogs, or data for decision support, where the most current data is not essential and the data is used as read-only.

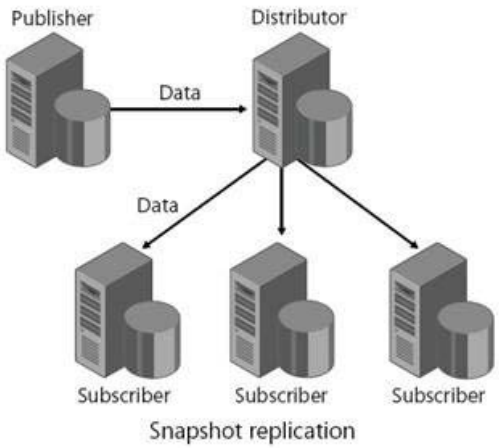

Fig.1. Snapshot Replication

Snapshot replication is helpful when:

- Data is mostly static and does not change often.
- It is acceptable to have copies of data that are out of date for a period of time.
- Replicating small volumes of data in which an entire refresh of the data is reasonable.

6.2 Merging replication

Merge replication is the process of distributing data from Publisher to Subscribers, allowing the Publisher and Subscribers to make updates while connected or disconnected, and then merging the updates between sites when they are connected.

Merge replication allows various sites to work autonomously and at a later time merge updates into a single, uniform result.

The initial snapshot is applied to Subscribers, and then changes are tracked to published data at the Publisher and at the Subscribers. The data is synchronized between servers continuously, at a scheduled time, or on demand. Because updates are made at more than one server, the same data may have been updated by the Publisher or by more than one Subscriber. Therefore, conflicts can occur when updates are merged.

Merge replication includes default and custom choices for conflict resolution that you can define as you configure a merge publication. When a conflict occurs, a

resolver is invoked by the Merge Agent and determines which data will be accepted and propagated to other sites.

Merge Replication is helpful when:

- Multiple Subscribers need to update data at various times and propagate those changes to the Publisher and to other Subscribers.
- Subscribers need to receive data, make changes offline, and later synchronize changes with the Publisher and other Subscribers.
- You do not expect many conflicts when data is updated at multiple sites (because the data is filtered into partitions and then published to different Subscribers or because of the uses of your application). However, if conflicts do occur, violations of ACID properties are acceptable.[1]

6.3 Transactional replication

In what could be considered the opposite of snapshot replication, transactional replication works by sending changes to the subscriber as they happen.

As an example, SQL Server processes all actions within the database using Transact-SQL statements. Each completed statement is called a *transaction*.

In transactional replication, each committed transaction is replicated to the subscriber *as it occurs*. You can control the replication process so that it will accumulate transactions and send them at timed intervals, or transmit all changes as they occur. You use this type of replication in environments having a lower degree of latency and higher bandwidth connections. Transactional replication requires a continuous and reliable connection, because the Transaction Log will grow quickly if the server is unable to connect for replication and might become unmanageable.

Transactional replication begins with a snapshot that sets up the initial copy. That copy is then later updated by the copied transactions. You can choose how often to update the snapshot, or choose not to update the snapshot after the first copy.

Once the initial snapshot has been copied, transactional replication uses the Log Reader agent to read the Transaction Log of the published database and stores new transactions in the DISTRIBUTION Database. The Distribution agent then transfers the transactions from the publisher to the subscriber.

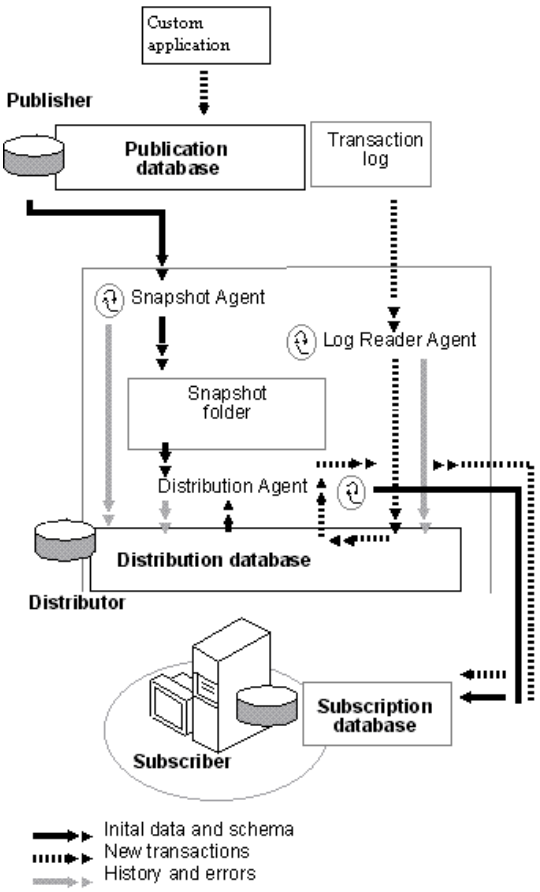

Fig.2. How it works: Transactional Replication [1]

Transactional replication with updating subscribers

An offshoot of standard transactional replication, this method of replication basically works the same way, but adds to subscribers the ability to update data. When a subscriber makes a change to data locally, SQL Server uses the Microsoft Distributed Transaction Coordinator (MSDTC), a component included with SQL Server 2000, to execute the same transaction on the

publisher. This process allows for replication scenarios in which the published data is considered read-only most of the time, but can be changed at the subscriber on occasion if needed. Transactional replication with updating subscribers requires a permanent and reliable connection of medium to high bandwidth. [5]

Transactional replication is helpful when:

- You want incremental changes to be propagated to Subscribers as they occur.
- You need transactions to adhere to ACID properties.
- Subscribers are reliably and/or frequently connected to the Publisher.[6]

7 Conclusions

It is obvious that Database Replication it's not a very simple process but if applied in the right circumstances it can be an extraordinary solution for developing better applications, for improving performance and for better experience for users.

These advantages do not come without a cost. Data replication obviously requires more storage, and updating replicated data can take more processing time than updating a single object.

In the same time, Database Replication can turn out to be complicated when it increases in size and magnitude but used properly, replication can improve considerably your data infrastructure.

Clients at the site to which the data is replicated experience improved performance because those clients can access data locally rather than connecting to a remote database server over a network and clients at all sites experience improved availability of replicated data. If the local copy of the replicated data is unavailable, clients can still access the remote copy of the data.

In a few words replication improves with availability and being highly distributed. Consider companies where users are disconnected during the day but need to update orders/inventories and other information automatically after normal working hours. Database Replication provides an easy solution when data must be highly distributed.

References

[1] Microsoft MSDN Library - http://msdn.microsoft.com
[2] WikiPedia - http://www.wikipedia.com
[3] Community for sharing technology information - http://www.tech-faq.com/
[4] IT information, Introduction to Database Replication- http://www.informit.com
[5] Mark A. Linsenbardt, Shane Stigler "McGraw-Hill/Osborne Media book SQL Server 2000 Administration" - Chapter 10, 'Replication'
[6] Sql Server Library TechNet – Microsoft - http://technet.microsoft.com

Marius Cristian MAZILU graduated from the Faculty of Cybernetics, Statistics and Economic Informatics of the Academy of Economic Studies in 2008. He got his Master Degree in the Database support for businesses department of the Academy of Economic Studies in 2010. He is currently attending the PhD program of the Academy of Economic Studies in the field of Economic Informatics. His domains of work are: Development and Management of Database Applications, Developing Web Applications for Businesses and Web Marketing.

Integration of Web Technologies in Software Applications. Is Web 2.0 a Solution?

Cezar Liviu CERVINSCHI, Diana BUTUCEA
Academy of Economic Studies, Bucharest, Romania
cezar@symbolsoftware.ro, dianabutucea@gmail.com

Starting from the idea that Web 2.0 represents "the era of dynamic web", the paper proposes to provide arguments (demonstrated by physical results) regarding the question that is at the foundation if this article. Due to the findings we can definitely affirm that Web 2.0 is a solution to building powerful and robust software, since the Internet has become more than just a simple presence on the users' desktop that develops easy access to information, services, entertainment, online transactions, e-commerce, e-learning and so on, but basically every kind of human or institutional interaction can happen online. This paper seeks to study the impact of two of these braches upon the user – e-commerce and e-testing. The statistic reports will be made on different sets of people, while the conclusions are the results of a detailed research and study of the applications' behaviour in the actual operating environment.
Keywords: *Web 2.0, web technology, e-commerce, online transaction, e-testing*

1 Introduction

The answer for the question that makes the object of this paper, "*Is Web 2.0 a solution?*", requires a multi-directional approach. While 'Web 1.0' is the synonym for "*the static web*", 'Web 2.0' represents the dynamic way of creating, developing and using applications over the Internet. Extremely user friendly interfaces with full content management options are developed quickly together with complex applications that have various functionalities and require a large and sophisticated amount of resources (especially regarding the security issue). Web development technologies are constantly improving their performance an offered facilities to the end-user: next to the expansion of the dedicated development frameworks, the hardware infrastructure that keeps up with the software also gets better by the day (an example of such a technology is *cloud-computing*). This allows web technologies together with these extended capacities, to run smoothly on a reliable logistic base without any technical issues. Even tough, for the moment, the hardware resources are maybe a bit too much for micro level implemented web applications (inside a financial institution or inside a series of interconnected financial institutions, for example), these resources must be considered at the time when a classic web server will become insufficient.

The "*Web 2.0*" term has first been quoted by Darcy DiNucci, in 1999, when, in her article, "*Fragmented future*" [5], [6], she writes: "*The Web we know now, which loads into a browser window in essentially static screenfulls, is only an embryo of the Web to come. The first glimmerings of Web 2.0 are beginning to appear, and we are just starting to see how that embryo might develop.*". Later on, John Robb [7] says that "*[Web 2.0] is a system that breaks with the old model of centralized Web sites and moves the power of the Web/Internet to the desktop.*".

In 2004, when the first official conference on Web 2.0 took place, "*O'Reilly Media and MediaLive*", the term became unanimously accepted. Consisting of everything that "*Web as Platform*" means, a generic term (which defines web-based software applications) maybe having even better functionalities then these. According to D. Best [4], the characteristics of Web 2.0 are rich user experience, user participation, dynamic content, metadata, web standards and scalability. Thus, we state that, the main phrase, when referring to Web 2.0, is

"*interaction*". We state this because, obviously, over the last few years, the web has become one of the most powerful means of information (in most of the cases, the most powerful), personal development mechanism, knowledge enrichment tool and so on. If ten years ago the Internet was just a sum of plain, static web pages, almost irrelevant for the user, nowadays the access to information is extremely simple and can even be enriched by editing the pages while browsing (e.g. Wikipedia). This way, the Web 2.0 technologies encourages and amplifies the user's role in the browsing experience:

- Instant Messaging (Yahoo! Messenger, GTalk): text-based, real time communication channel between two or more users;
- Internet Telephony (Skype): real time audio communication between two or more users;
- Blogs (WordPress, Blogger): easily customizable web pages, where users can post topics and others can express their opinion and comment upon;
- Social Networks (Facebook, Twitter): web applications that create links between users that share the same interests, hobbies or other;
- Wikis (Wikipedia): web applications that allow creating and/or editing web page content, mostly used for developing website communities that are able to rapidly store massive amount of information.

What this article aims, is to present practical results obtained by interviewing users about how the new Web technologies in general, e-commerce and e-testing, in particular, were seen by them after using these software platforms.

2 Concepts and technologies

2.1. Concepts

From a minimalistic look, the Web 2.0 can be described throughout three major components:

- Service Oriented Architecture (SOA) - defines the structure of web applications developed using the 2.0 model by implementing the functionality of mutual inclusion of the Web pages' contents (e.g. Web services);
- Rich Internet Application (RIA) – defines the implementation of the classic software packages facilities, developed and included inside the Internet environment (the web browser);
- Social Web - defines the user as part of the web development process.

As shown above, the three listed components define the client - server architectural model, and include the concepts of client-side software, server-side software, content uniformity and use of network protocols. Newer versions of browsers have included in their standard software extensions or plug-ins that allow interaction with web 2.0 elements, elements that were missing in web 1.0 (XML, RSS, AJAX and others).

Other important concepts defined by the literature [1] are SLATES - Search, Links, Authoring, Tags, Extensions, Signals.

From a functional point of view, the available web technologies are divided into two main categories: server-side web technologies and client-side web technologies.

2.2. Server-side web technologies

Given the extremely rapid development of the Internet, new programming languages have emerged and distinguished themselves as powerful tools in order to standardize the content of web pages. Thus, languages like PHP, JSP, ASP, Ruby and others have functions and features that facilitate working with XML, RSS or JSON files, making the communication between two or more web applications possible, even if they were initially developed in different programming languages.

2.3. Client-side web technologies

Regarding the Web 2.0 technologies used on the client-side, these aimed to develop tools that facilitate the creation of dynamic, easily understandable and usable interfaces but, especially, to create visual and functional effects. The AJAX (Asynchronous JavaScript and XML), Adobe Flash, Adobe Flex and JavaScript framework and libraries (Yahoo User Interface library, jQuery, MooTools) technologies have been developed.

Each of the listed technologies have special features. The developer's decision of using one over another remains at his will, his work experience and the purpose of the software application. While the AJAX technology offers facilities for working with asynchronous requests to the server (the requests are running in the background of the browser), Flex and Flash technologies are capable of displaying large amounts of information in video and audio formats. However, essentially, what a technology has to offer better than another is only a small portion of the package, namely to work with JavaScript or DOM (Data Object Model) objects.

3. Web technologies integrated in e-commerce solutions

Electronic commerce (e-commerce) has been, since the early expansion of the Internet, a major challenge for developers. Problems like the connection between the parties, security of connection, security of transaction, the implemented mechanism, the participants in the business process all were the subject of many debates and development projects. As for the solution to these problems, the development of e-commerce solutions can be divided into four stages: the pre-web stage, the reactive web stage, the interactive web stage and the integrative web stage [2].

Each of the four stages presents special features regarding the development of the Internet, the technologies that were used and developed becoming more advanced. Table 1 summarizes the stages of the e-commerce applications.

Table 1. Evolution of the Internet

	Pre Web
Period	Before 1990
Participants	One-to-one
Work environment	Dedicated connection
Communication	FTP, telnet, EDI
Presentation / Representation	ASCII
Programming language	Fortran, C, C++, Perl
Storage	SQL, DbaseIII
	Reactive Web
Period	Early '90s
Participants	One-to-one
Work environment	Web page, portal, company
Communication	HTTP, CGI
Presentation / Representation	HTML
Programming language	C, C++, Perl
Storage	Access, DbaseIV, Oracle, SQL
	Interactive Web
Period	Mid '90s
Participants	One-to-many
Work environment	Common market, trader, dealer, broker
Communication	SSL, Cookie
Presentation / Representation	SSI, VRML, Flash
Programming language	Java, PHP
Storage	ODBC
	Integrative Web
Period	Late '90s
Participants	Many-to-many
Work environment	Platform, community, business, industry
Communication	Wap, PKI
Presentation / Representation	XML, XHTML
Programming language	ASP, JSP, JDBC, ColdFusion, SQLJ, JavaBeans
Storage	JDBC, SQLJ

The experiment conducted towards finding an answer to the question "Is Web 2.0 a solution for e-commerce?" has been made on a sample of 20 people with ages between 20 and 45. The issue was whether, by visiting two online stores, they became convinced to order products online. The two stores have been developed using the PHP server-side technology and the AJAX client-side technology (Yahoo User Interface libraries, MooTools, jQuery). The ergonomics of the two software platforms and its role regarding the customer's decision to purchase the products offered for sale has also been tested. Appendix 1 presents the evaluation questionnaire together with the achieved scores and the metric chosen for the performed reporting.

The results of the study reveal that, for the first online store, which, on a scale of 1 to 10, has a 9.51 average in appearance and an 8.46 average on the traded products' utility, 74.6% of the customers have shown an interest in purchasing products from this store. For the second online store, which received an average of 7.34 for appearance and 6.57 for utility, the results reveal a strong correlation between these indicators and the customers' interest in purchasing the products it sells, the percentage of 47.3% of the subjects interested in buying was clearly lower than the one for the first store.

4. Web technologies integrated in e-testing platforms

Software platforms for computer-assisted learning and testing represents a special class of web applications, mainly due to different possible modes of development, integration and implementation. The above technologies are capable of developing modular platforms, easily to improve, easily to manipulate and that creates easy access to data, reports and result graphs.

A current trend in computer-assisted learning and testing processes is the standardization of the structure of these software systems. Thus, international organizations (many of them united under a single aegis), such as IMS - Instructional Management Systems, LTSCALE - Learning Technology Standards Committees, ADL - Advanced Distributed Learning, decided to create a permanent collaboration in e-learning standards . The most common such standards are based on metadata: SCORM and Learning Design[8]. Given the direction of standardization based on metadata the format in which the questions will be integrated into the system must be clearly defined. Thus, they can be of various types (with short, single or multiple answer, with free answer, with adding item answer and so on) and raises two issues - how will the questions be included into the application and the method or algorithm through whom they will be added into the evaluation tests. Regarding the first problem, research is conducted in the field of automation for the training process (for example, linking the taught material with the assessed one through semantic networks). Regarding the tests generation mechanism, there have been created linear test generating algorithms (the same set of questions for all those assessed), dynamic test generating algorithms (different sets of questions) or adaptive test generating algorithms (dependent on the knowledge of the assessed person, iterative [3]).

The experimental research to find an answer for the question "Is Web 2.0 a solution for online testing?" was conducted on a sample of 20 students aged between 20 and 23. The issue was that, after taking three tests using an e-testing software platform, the subjects would complete a questionnaire on efficiency, ergonomics and functionality of the web application. The platform was developed using PHP server-side technology and AJAX client-side technology (YUI libraries, MooTools, jQuery). The questionnaire, the scores and the chosen metrics defined for the reports are presented in Appendix 2.

The experimental results show that 82.5% of the subjects have rapidly adapted to the interface and functions of the

platform, which is quoted by most of them as having a high level of usability. A correlation between the answer regarding the ergonomics and the tests' scores can be seen, subjects with less good scores defining the application less ergonomic and difficult to use, while the subjects with high scores are those who have rapidly adapted to the interface. Regarding the utility of the application, 79.6% of the subjects consider it a useful tool and are wiling to help on improving it by providing comments and suggestions upon it.

5. Conclusions

The present paper wants to argue, both theoretically and experimentally, that the answer to the main issue here - "Is Web 2.0 a solution in software development?". Given the results of the study for the two areas of research, electronic commerce and computer-aided testing, we can strongly say that Web 2.0 is a solution. The currently available technologies on the market and their possibilities of development, both server-side and client-side consist of valuable resources for the development of Web 2.0 and gradual transition towards Web 3.0.

Correlations between aesthetics and the desire to buy the product, on the one hand, usability and the testing's results, on the other hand, emphasizes that a good design of the user-interface creates the premises of having a rich and positive experience in interaction with the software platform, positive results increasing with the aesthetic level of the interface.

At the same time, from a functional point of view, it has been proven that the choice of implementing the software platforms using PHP as server-side technology and AJAX (YUI libraries, MooTools, JQuery) as client-side technology, has been a viable solution for the user, the impact of visual and functional effects of the applications being positive, as shown by the statistics.

Essentially, Web 2.0 is not only users interacting with each other through all means possible (instant messaging, blogs, internet telephony, social networking or others), but especially creating an more attractive environment (aesthetically and functionally) for the user, improving the browsing experience and, obviously, convincing the user to return to the web application.

6. Acknowledgments

This work was cofinanced from the European Social Fund through Sectorial Operational Programme Human Resources Development 2007-2013, project number POSDRU/107/1.5/S/77213 „Ph.D. for a career in interdisciplinary economic research at the European standards" (DOCCENT).

References

[1] McAfee, A., „Enterprise 2.0: The Dawn of Emergent Collaboration." *MIT Sloan Management review*, Vol. 47, No. 3, 2006, pp. 21–28.

[2] Sung-Chi Chu, Lawrence C. Leung, Yer Van Hui, Waiman Cheung, „Evolution of e-commerce Web sites: A conceptual framework and a longitudinal study", *Information & Management*, Vol. 44, 2007, pp. 154–164

[3] Armenski Goce, S., Gusev, M., „Infrastructure for e-Testing", *Facta Universitatis*, *Ser. Elec. Energ.*, Vol. 18, No. 2, 2005, pp. 181-204

[4] Best, D., „Web 2.0 Next Big Thing or Next Big Internet Bubble?", *Lecture Web Information Systems. Technische Universiteit Eindhoven*, 2006

[5] DiNucci, D., „*Fragmented Future*", *Print* 53 (4): 32, 1999

[6] Who coined Web 2.0? : Darcy DiNucci, http://www.cole20.com/who-coined-web-20-darcy-dinucci/

[7] Web 2.0, John Robb's Weblog, http://jrobb.mindplex.org/2003/08/16.html

Cezar Liviu CERVINSCHI graduated from the Faculty of Automatics and Applied Informatics of the Transilvania University of Brasov in 2009. With a masters degree in Information Technology, his interest in web technologies, web development, integrated software systems and database administration brought the motivation for advanced research in this field of science, since his main target is building robust, reliable and complex software solutions. At present, he is a PhD candidate at the Academy of Economic Studies in Bucharest at the Economic Informatics department studying web technologies, databases, online security, e-business and e-marketing.

Diana BUTUCEA graduated in 2008 from the Economic International Relations Department of the Academy of Economic Studies in Bucharest, in 2009 from the Faculty of Automatics and Applied Informatics of the Transilvania University of Brasov and in 2010 from the Economic Informatics masters at the Academy of Economic Studies in Bucharest. Her parallel interests, in economy and software engineering, are now merging into her studies and researches since she is PhD candidate at the Academy of Economic Studies, Bucharest, studying integrated software systems, web technologies and e-learning platforms.

Commercially Available Data Mining Tools used in the Economic Environment

Mihai ANDRONIE[1], Daniel CRIȘAN[2]
[1]Academy of Economic Studies, Bucharest, Romania
[2]Eidgenössische Technische Hochschule Zürich, Switzerland
mihai_a380@yahoo.com, crisand@student.ethz.ch

This paper presents some of the most common commercially available data mining tools, with their most important features, side by side, and some considerations regarding the evaluation of data mining tools by companies that want to acquire such a system. Among some of the most important factors that a company has to take into account are the amounts of data available, how it is stored and the data mining tasks that must be performed, but there are also others. Not the last it should be mentioned that the cost of a data mining system is important for a company, having a limiting effect on the expansion of the data mining products market towards small companies.
Keywords: *Data Mining, Data Analysis, Information System, Data Mining Tools*

1 Introduction

Following the significant advances in the information technology during the last decades, our society faced an increase of the volume of data produced and stored in various fields of activity. For this reason there were developed various advanced technologies to process the huge volumes of data suggestively called data mining techniques (an analogy to the mining processes that are carried on to extract precious metals from the ground).

Data mining answers the need to process huge volumes of data to gain useful information and knowledge in various fields such as market analysis, production control, marketing, scientific research and others.

In the economic field there are many benefits of using data mining techniques in integrated information systems. In the following there will be mentioned some of the most important techniques used in conjunction with their range of applicability in the economy.

There will also be presented some of the most representative commercial available data mining tools with their most important features, side by side so that their key features to be outlined to the reader.

Not the least important of the material presented in this paper are the factors that a company should take into account before deciding to buy such a commercially available data mining tool, in order to use it for its management to improve its decision making capabilities.

Besides using data mining techniques for company management support, among the most useful applications of data mining in economics is the area of retail distribution of merchandise. There exist traditional or online stores that record information in their databases regarding their sales, customers, etc. All this information gathered over a long stretch of time can be analyzed using data mining techniques.

Data mining tools can be used even for the improvement of a company's information system, this having measurable economic consequences even if it isn't an economic purpose in its own.

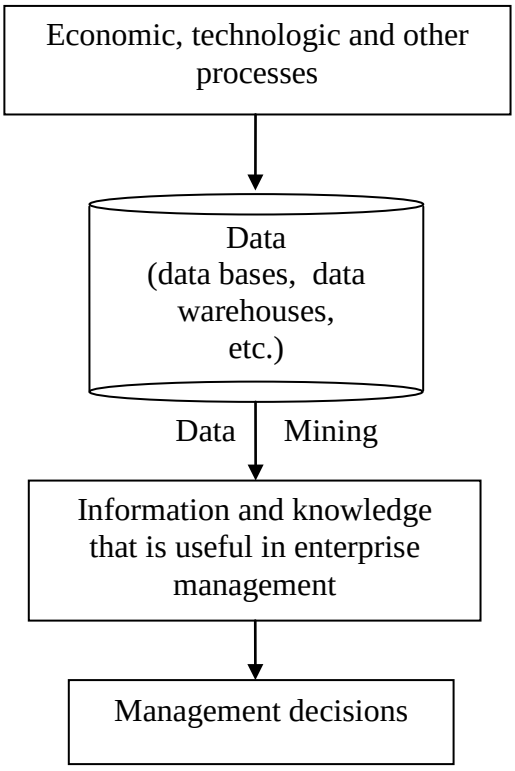

Fig. 1. The role of the data mining techniques in taking management decisions

We have for the cases previously described different applications of the data mining techniques [4] such as:

- The well known marketing problem [3] of shopping basket analysis from which there can be obtained different relationships between the products which were sold together in the past;
- Using data mining techniques to assist in designing data warehouses for a specific application. In this way one can more easily determine what dimensions should be included in such a model, according to the data available;
- Determining the effectiveness of sales campaigns – it is another economic problem to which a solution can be found using different data mining techniques; if it is found that sales campaigns do not have the desired results these can be improved or canceled;
- Analysis of customer behavior; customers may or may not be loyal to the company and they may respond more or less on certain factors which depend on the company's policy; knowledge of the customer's behavior can be very useful in adopting a business strategy to maximize profits.
- Another very important application of data mining in economics is to use advanced data analysis techniques for strategic management of enterprises. For a company is vitally important that decisions that are taken by the top management may be taken on an informed basis and not based solely on the talent and experience of the manager. This application of data mining techniques became possible by making predictions based on the data that the company has access to (data from its own databases or even external sources of data). This is perhaps the area where data mining techniques are the most important because the data gathered in time cannot be managed and analyzed traditionally being in huge quantities.

2. Factors that influenced the development of data mining techniques

Early in the development of data analysis there have been various statistical techniques for extracting useful knowledge from data. These techniques didn't require a computing power similar with what we encounter today.

The first data mining techniques were derived from statistical methods and are called classical techniques (such as statistical analysis or cluster analysis which are even today among the data mining tasks performed by modern data mining systems).

The classical techniques were originally associated with data collections and were later adapted to analyze huge volumes of data. They remain the most used data mining techniques because specialists are very familiar with them, being usually applied on databases and data warehouses.

Once the technology has evolved there appeared more diverse data types, so that new generation data mining techniques (also called next generation techniques) started to

be developed.

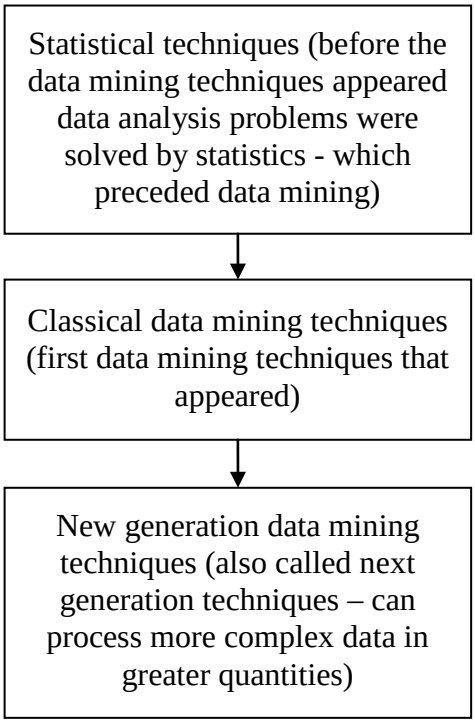

Fig. 2. The development of data mining techniques since their beginning

New generation data mining techniques are able to process a much larger volume of data (compared to the first databases and data warehouses) in a relatively much shorter period of time. Also, these techniques can easily process the unstructured data types that may prove a valuable source of data but are little exploited.

Taking into account the current evolutions in the data mining field we can anticipate that in the near future it will be possible to exploit data such as the content of papers that until now were mostly on printed format like books (scanned prior to being put into electronic format), maps, and others.

Also we expect that the Internet will become one of the most important sources of data publicly available and it will be exploited at levels not seen until now. In this case the only major obstacles that have to be overcome are the high degree of heterogeneity of web-based data and choosing the relevant information from the millions of web pages that are available. Not the least it should be taken into consideration that not all the information found on the Internet is correct, the analysis of incorrect data leading to errors in the results.

Last but not least it should be mentioned that it is not conceivable an advanced data analysis without a prior development of effective methods of data storage, retrieval and processing, this being facilitated by technologies in the fields of databases and data collections. For this reason it is not a coincidence that most of the data mining performed until today had as basis relational databases so relational databases came quickly to be a preferred option for storing and managing large volumes of data. Among the factors that contributed to the development, acceptance and maturity of relational database technologies must be mentioned online transaction processing databases (OLTP).

3. The role of data mining techniques

Although the field of data mining is relatively recent, as a result of the the research in the last years have appeared a large variety of algorithms, methods and techniques which allow users to perform a wide range of functions using these techniques. These functions are important for an enterprise that has to choose a data mining system to implement so that they will be mentioned below.

Data mining techniques, data mining algorithms and data mining methods each have a purpose for which they were designed, performing a specific data mining task [5]. Among the most common tasks the data mining systems perform include clustering, association, classification, regression, forecasting, analysis of data sequences (series of discrete values) or outlier analysis (analysis of exceptions) (Fig. 3.).

A data mining system can be specialized for one task or can be general enough to perform multiple data mining tasks. Most of the available data mining tools described below implement a number of algorithms

and are able to perform most of these tasks, being general enough to cover the needs of the customers.

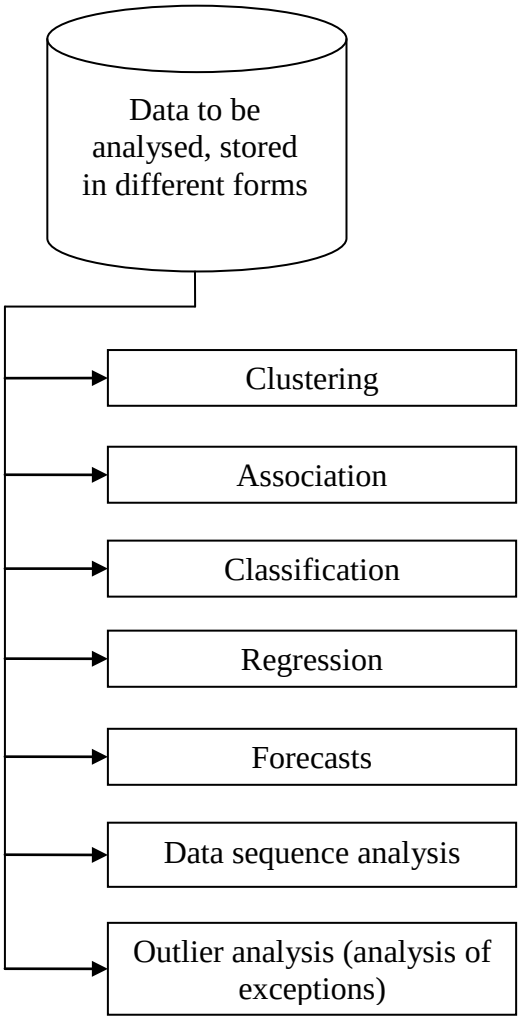

Fig. 3. The main data mining tasks that can be used to analyze a large amount of data

In the next paragraphs will be summarized these data mining tasks and will be highlighted the specific characteristics for each of them. Some of them can be used to some extent for similar problems, but there are also notable differences that distinguish one from another [6]:

- *Association* – aims to identify the most common sets of "objects" that appear together, sometimes the end user can choose which objects he wants to be analyzed or how frequent he wants them to be; the user can also set rules of association to better control the analysis; Among the most common applications of association are the identification of products most frequently sold together (market basket analysis), identifying areas where more than two products are sold together most often or the identification of time periods during which the sales are growing;
- *Classification* – It is one of the most common data mining tasks, responding to issues such as risk analysis; unlike clustering, where categories are not predetermined, the objects are inserted into categories, called classes, according to predetermined characteristics; classification algorithms need to work in a supervised mode, requiring some criteria in order to categorize objects; for this reason usually there must be previously established criteria for classification which can be obtained as a result of the analysis of historical data held; some classification techniques include decision trees, neural networks, etc. As examples can be mentioned the classification of bank customers in terms of how they pay debts, insurance risk analysis in order to determine the insurance premium for various specific cases or establishing property taxes based on certain criteria (value, area, size);
- *Clustering* – helps to find categories of "objects" based on their attributes; groups are formed containing "objects" similar in many respects (having similar attributes); a difference between clustering and classification is that clustering algorithms can work in an unsupervised mode, taking as input the attributes of the objects and offering as outputs the groups obtained (clusters); Two examples of clustering are grouping the customers of a company according to their purchase value and grouping available products by their features;
- *Forecast* – aims to make a prediction while taking into account past values usually in the form of series of events conducted over time and unlike regression can take into account other factors such as periodical fluctuations of events; As examples of forecasting can be mentioned the seasonal

forecasts in different fields of activity (if the variation is cyclical the regression is not suitable for analysis) or the forecast of sales of goods for the future based on the data available from the past;

- *Outlier analysis* – this type of analysis is intended to detect a number of "objects" that behave very differently from the rest; as an example it is most frequently used for fraud detection, intrusion detection or finding errors;
- *Regression* – is a data mining task that has its origin in statistics where it was used extensively; it resembles from some points of view with classification the difference being the fact that it works on continuous-valued attributes; regression can be used for predictions also, when the data analyzed is time related, but not for cyclic predictions (linear or polynomial non periodic predictions); a frequent use of regression is the calculation of intermediate values by interpolation;
- *Sequence analysis* – it is used to identify patterns in a series of discrete values; the main difference between sequence analysis and association is that the first is searching for the order of events and the transitions between different states while the second only searches for correlations between supposed independent objects; an example of sequence analysis can be the identification of models followed by a company's sales over time.

As it can be seen from those stated before, there are some similarities between some of the data mining tasks. Thus for example it can be said that both the classification and the clustering have the effect of dividing "objects" into groups or classes, but the difference between them is how this action is done. The classification deals with pre-existing classes that are defined before the analysis is performed, while the clustering groups "objects" into groups which are constructed taking into account the characteristics of "objects" under consideration.

Another difference worth mentioning is that between association and sequence analysis. If the first simply notes some rules and patterns in the data analyzed, the latter is looking for similar models on some data ordered in time, trying to determine a way in which these models are ordered, assuming that there are some correlations between the events.

It also should be noted the difference between the tasks of regression and forecasting. Regression plots a general trend, if carried out in time, can be used among others to make simple forecasts also. However, the forecast task is more complex and can take into account other factors, working on cyclical events, etc. For example forecasts can be used for data that varies from a period of time to another like sales around the seasons of the year.

A correlation between different data mining tasks that is worth mentioning is that between outlier analysis and other tasks like clustering or classification. The latter two are not intended to find exceptions in the data available (outliers) but can also be used for these tasks because an exception can be classified in a different class (or cannot be attributed to a certain class at all) and it can also form an individual cluster, very different from the others. Anyway, it is not needed to use classification and clustering to analyze exceptions because there are specialized techniques and algorithms for this task.

4. Considerations Regarding the Evaluation of Data Mining Tools

For a system with embedded data mining techniques to be profitable for a company's business it is necessary for it to have certain features. The characteristics that a computer system with embedded data mining techniques must have should be in connection with the business where it needs to be used and its requirements.

First of all, for a company operating in the economic field, whatever its domain of activity is, it is necessary to have an existing

computer system with data to be analyzed prior to the data analysis. Only after this requirement is met, a data mining system can be installed and used to extract useful information from the available data.

Another fact that should be considered is that in many cases a company's computer system consists of a collection of applications that are interconnected and have access to data that is often stored in relational databases. The users of the enterprise's computer system, usually employees of the company, have access to this data using these applications, not directly (as shown in Fig. 4). To integrate a data mining system in the computer system of a company it must be able to retrieve data from the computer system's database using their analysis techniques and performing tasks such as clustering, association, prediction, regression, etc. [3]. Following this analysis on the existing data there can be extracted useful information and knowledge that can be used at various levels within the company, from the basic business processes to the top management for decision making.

Another thing to consider before deciding to integrate a data mining system in a company's computer system is the importance of the data that the company has. The data that the company owns or has access to must be relevant and contain potential knowledge to be searched for using data mining techniques. For this it can be made an assessment of the data at which a company has access. In most cases the use of data mining techniques brings benefits to companies, regardless of their field of activity.

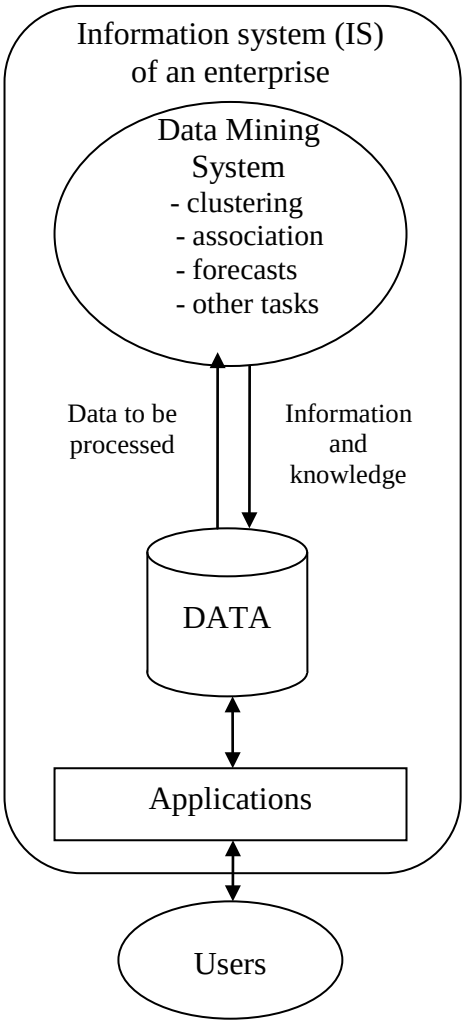

Fig. 4. The integration of a data mining system into an enterprise's information system

Not the least important it must be taken into account the costs of implementing a data mining system. For small companies the high costs of data mining products can be an important factor that can slow down the penetration of data mining tools on the market. On the other hand for large companies which are more financially powerful, the use of these products has already shown benefits.

Companies nowadays are provided with a large variety of data mining commercially available tools. The only problems they face is identifying their needs and evaluate the available tools in order to choose the data mining system that best fits its needs and budget.

Because the data mining products available

on the market tend to have different features in most of their aspects, a company should plan to consider a number of criteria against which to evaluate a data mining tool before deciding which data mining system to acquire [4]. Among the most important criteria to be considered are:

- The amount of data that is available to be analyzed. Is it necessary to buy a more powerful data mining tool which is more expensive, or it is enough something simpler?
- The amount of preprocessing the available data needs before being mined. If the data is stored in relational databases it is easier to analyze it, and most of the data mining systems will work. But if for example the data is printed text, first it must be scanned or introduced in the computer and only after it should be analyzed using a data mining tool that can take text as an input.
- The way the available data is stored. If the data is stored in relational databases it is needed a data mining system that can work on databases, but if the data comes from large data streams it is necessary a more specialized data mining tool that can make real time analysis.
- How complex the analysis must be. For simpler analysis the data mining system can be a more affordable one but for more complex analysis a specialized data mining system is necessary.
- What data mining tasks the company wants to be performed by the data mining system: association, clustering, classification, outlier analysis, regression, forecasts or others.
- The data mining system that is chosen must cover all the future needs of the company. If it is intended to make other types of analysis in the future those bust be covered too by the system.
- Flexibility. A data mining system must be flexible to be adapted to different types of analysis. For each data mining task there can be implemented more than one data analysis algorithm.
- The coupling between the data mining system and the database management system that the company is using (in the case of analyzing data stored in databases). A data mining system that is more coupled with a database management system has access to its internal functions and is more efficient. A data mining system that isn't coupled with a database management system uses its external functions to access data (like SQL queries for example) and for this reason is not so effective, but it is more flexible.
- API interfaces available. Some data mining systems offer API function libraries that make possible the integration of data mining functions in the software that a company is already using. This is a great advantage because it eliminates the need of running more applications at the same time, one for current use, and one for data analysis.
- Scalability of the system – it is very important if the company's database extends and becomes necessary to analyze a data volume higher than originally planned.
- How user friendly a data mining system is. It is important because most of the times the persons using the system are not IT specialists. Visualization tools are important because they make the presentation of the obtained results to be more suitable to be grasped by the human end user.

5. Features of Some Commercially Available Data Mining Tools

Data mining systems available on the market are usually the products of companies coming from the databases, hardware, statistical analysis or other related fields. Some of the most popular products on the market are presented in Tab. 1 [6]. where there are tresented the main features of each. Among these there are included: Intelligent Miner, produced by IBM, the data mining tools included in Microsoft SQL Server 2005 package and later, Oracle Data Mining, working with Oracle 10g database, and

others. Each of these solutions implement several data mining functions, each function in turn using several techniques and methods of performing the data analysis.

Table 1. Some data mining products commercially available

Data Mining Product	**Main Features**
SAS Enterprise Miner	• It comes from statistics; • Easy to use graphical interface; • Rich set of algorithms including algorithms for data mining: decision trees, neural networks, regression, association, etc. • Ability to analyze text.
SPSS	• It comes from statistics; • Includes among others, decision tree data mining algorithms (Answer Tree); • Allows users to perform data cleansing and data transformation.
IBM Intelligent Miner	• It comes from the database field; • Features advanced visualization tools and data presentation; • Compatible with PMML language (Predictive Modeling Markup Language) for exporting the data models found; • Can work with DB2 database management system.
Microsoft SQL Server 2005	• It comes from the database field of activity; • Among others, it offers algorithms for decision trees, prediction and clustering; • Implements the OLE DB standard for Data Mining, which defines a data mining language similar to SQL; • Features an easy to use API interface for facilitating the integration of data mining facilities into the user applications.
Oracle Data Mining (from Oracle 10g)	• It comes from the database; • It started with algorithms such as association and Naive Bayes (version 9i) and with the 10g version it includes a great variety of algorithms; • Integrates Java Data Mining API, a Java package for including the data mining facilities into the user's applications.
Angoss Knowledge STUDIO	• Presents algorithms for building decision trees, cluster analysis (grouping) and predictive models; • Allows users to exploit data in different forms; • Offers powerful visualization tools of the results that make it very user friendly; • It is compatible with other databases such as Microsoft SQL Server, and can interact with them at datamining level.
KXEN	• Has algorithms for regression, time series analysis, classification, etc. • Implements procedures for working with OLAP data cubes; • It can retrieve data from spreadsheet programs like Microsoft Excel.

By analysing the previous table it can be seen that each product described has different features and for this reason for a certain case of a company it has to find

which one best meets its requirements.

6. Data mining tools used in the business environment

A study realized by Daniel Andersson and Hannes Fries [1] tries to analyze if companies use data mining tools to improve decision-making capacity. The study presented here is based on a series of 95 large companies from Sweden, that are chosen at random in order to see how many of them use a data mining system and how their managers see it.

The results of the study show that only 30% of the analyzed companies used a data mining system at that time (Fig. 5.) while more than 50% of their managers were seeking with interest to the developments in this area and intended to acquire such a solution in the near future (Fig. 6.). The same study predicted that future data mining systems market would grow by 100% over the next four years.

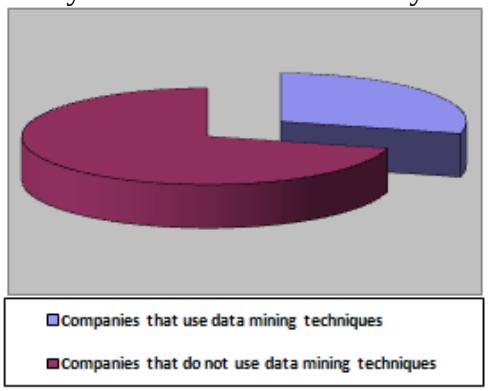

Fig. 5. Percentage of companies using data mining techniques

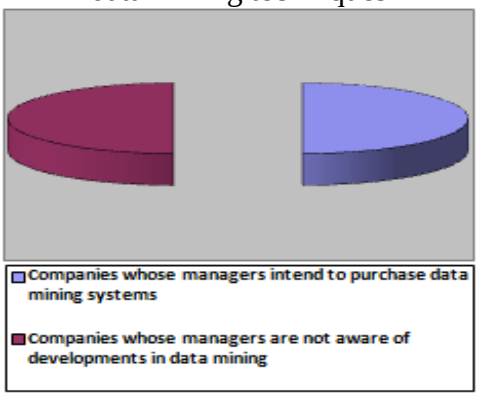

Fig. 6. Percentage of companies whose management is aware of developments in data mining and intend to implement such a system.

We believe that the results of this study can not be extended to other countries others than Sweden without taking into account the differences in technological development.

Therefore we can assume that in countries like those from Eastern Europe which are less technologicaly advanced the percentage of using data mining technology is much lower than the previously mentioned study indicates, but also the market potential could be greater. Also we must consider the fact that no small firms were considered in the study and that for many of them the use of data mining products is seen more as an option and not as something that would bring immediate profit. This leads us to believe that the data mining products market is still in its infancy and big developments in this domain can be expected during the next period of time [2]. Manufacturers of such systems will need to aim their products also towards small firms that have a great potential and are in an increased number this meaning they could be an important market.

7. Conclusions

For a business that wants to use a data mining system to optimize its activities there are available on the market various data mining commercial systems that can be integrated into that company's computer system, each of them having its own features.

By putting side by side different commercially available data mining products one can conclude that a company must first evaluate its needs regarding the data mining analysis and find the product that best covers them and fits the available budget.

There are a number of factors to be considered by a company before deciding which data mining system to use that are presented previously in this paper, like the amount of data available, the way the data is stored or the data mining tasks to be performed.

Because until recently only large companies had access to large volumes of data and financially afforded to use data mining tools,

the data mining market was mainly aimed to them. Despite this, as studies show, only a small percentage of the big companies use such tools to optimize their activities, but the managers are generally aware of the advantages brought by technology so it is expected that in the future the data mining tools market will grow.

The data mining tools market for small companies is not as developed as that for large companies but it has a great potential. The only things that are needed for its further development are the appearance of data mining tools affordable enough and the awareness of the small companies' managers.

References

[1] D. Andersson, H Fries: *Data Mining Maturity. A Quantitative Study of Large Companies in Sweden*, Jonkoping University, Master's Thesis in Informatics, 2008;

[2] M. ANDRONIE - *Modern data mining techniques*, The Ninth International Conference on Informatics in Economy IE, Bucureşti 2009, p. 753-757, ISBN 978-606-505-172-2;

[3] D. Hand, H. Mannila, P. Smith: *Principles of Data Mining*, Prentice Hall India, 2006;

[4] J. Han, M. Kamber: *Data Mining. Concepts and Techniques*, *Second Edition*, Elsevier, 2006;

[5] M. Andronie, M. ANDRONIE: *Analiza datelor stocate in depozite mari de date*, Sesiunea de comunicari stiintifice a cadrelor didactice din facultãtile economice ale Universitãtii Spiru Haret, Bucureşti, 2008, ISBN 978-973-163-230-8;

[6] Z. Hui Tang, J. MacLennan: *Data Mining with SQL Server 2005*, Wiley, 2005.

Mihai ANDRONIE is a graduate of the Faculty of Automatic Control and Computers, of the Politehnica University Bucharest in 2006. He is currently a PHD student at the Academy of Economic Studies of Bucharest. His domains of work are: informatics systems and databases. He participated as an author at the book "Administrarea bazelor de date" (Database management) (2008) and published several papers on domains like data mining, economic process optimization and others. He also participated at national and international conferences on his domain of activity.

Daniel CRIŞAN received a M.Sc. degree in Informatics from Ecole Polytechnique, France in 2008 and in Computer Science from EPF Lausanne, Switzerland in 2010. He is currently a PhD student in the Department of Information Technology and Electrical Engineering of ETH Zürich, Switzerland. His research interests include datacenter networking, routing and congestion management. He contributed to a novel routing scheme that efficiently exploits path diversity in datacenter networks. He co-authored a paper describing this method which was ranked #1 at IEEE Symposium on High-Performance Interconnects 2010.

Database Optimizing Services

Adrian GHENCEA[1], Immo GIEGER[2]
[1]University Titu Maiorescu Bucharest, Romania
[2]Bodenstedt-Wilhelmschule Peine, Deutschland
aghencea@yahoo.com

Almost every organization has at its centre a database. The database provides support for conducting different activities, whether it is production, sales and marketing or internal operations. Every day, a database is accessed for help in strategic decisions. The satisfaction therefore of such needs is entailed with a high quality security and availability.
Those needs can be realised using a DBMS (Database Management System) which is, in fact, software for a database. Technically speaking, it is software which uses a standard method of cataloguing, recovery, and running different data queries. DBMS manages the input data, organizes it, and provides ways of modifying or extracting the data by its users or other programs. Managing the database is an operation that requires periodical updates, optimizing and monitoring.
Keywords: *database, database management system (DBMS), indexing, optimizing, cost for optimized databases.*

1 Introduction

The purpose of the document is to present representative notions about basic optimizing for databases, using mathematical estimation for costs in different types of queries, a review of the level of attained performances, and the effects of different physical access structures in specific query examples. The target group should be familiar with SQL and basic concepts in relational databases. This way, execution strategies for complex queries can be made, allowing the use of knowledge for obtaining information at a lower cost. A database goes through a series of transformations until its final use, starting with *data modelling, database designing and development,* and ending with *its maintenance and optimization.*

2. Database modelling

Data modelling

The data model is more focused on the data that is required and the way in which those should be organised and less on the operations that will be made on the data. The data modelling stages involve the structure, the integrity, the manipulation and the query. There are multiple assets regarding this, such as:

1. Defining the way in which the data should be organised (hierarchical network, relational and object-focused). This provides a definition of rules that restrict what instances of the defined structure are allowed/premises.
2. Offers a data updating protocol.
3. Offers a method for data queries.

A simple structure of data communication which is easily understandable by the final user is the actual result of data modelling.

Customized databases/ Database development

The databases are developed and customized to answer the demands of the customer. The importance of custom databases is major because through them the commercialization of the products of services directly to the target customer becomes possible. The quality of a database is maintained through regular updates.

Database designing
If databases have any of the following problems: malfunction, insecure or inaccurate data or the database has degraded and lost its flexibility, then that is the moment for a new database.
Therefore, the types of specific data and the storage mechanisms have to be defined in order to ensure the integrity of the data through rules and mechanisms of correctly applying the operational principles. All databases should be constructed in regard to the specifications of the customers, including its user interface and functionality. Using the data by including them into a website is possible.

Data mining
Data mining is 'the science of extracting useful information from larger datasets and databases. Every organization wants its business and undergoing processes optimized for the best productivity. The business processes that require optimization include Customer Relationship Management (CRM), Quality Control, Prices and Delivery System etc. Data Mining is a data exploiting discipline which underlines the errors in those processes, using sophisticated algorithms. Data Mining is done on processed data and includes the analysis and determination of the mistakes.

Database Migration
Database Migration represents the transfer (or migration) of basic database schemes and data into the database management, such as Oracle, IBM DB2, MS-SQL Server, MySQL etc. There is a database migration system which allows the reliability and integrity of the data. The migration from one database platform can be difficult and time consuming because of the differences between the standards. However, quick migration of the data between different databases that ensures the integrity of the data is possible, without the loss of any data. Ensuring the access to the data and its protection is essential, especially when large quantities of data or important applications are being moved between systems. The security, availability and reliability of the data can be assured through providing experience in the projecting and applying the database infrastructure with oracle or Microsoft SQL server.

Database Maintenance
Database maintenance is a very important process in every organization. After a secure development of the database, the next process of major importance is the maintenance of the database, which offers an update, a backup and high security.

We could ask ourselves, why does a company need database maintenance?

When a database becomes altered, it is easily observed that the records no longer reflect the reality. This problem usually occurs in case of database deterioration. To remove any doubt regarding the integrity of the index a manual update and a regular backup is recommended.

As the activity of the organization grows, so does the dimension of the database. A useful practice is to periodically remove unusable data thus increasing the access to the database. Compression of the database will allow easier data supply and simplistic handling of the relevant information from the database.
The same database can be maintained in such way that it will offer the correct result for different questions. For example, the same discussion list can be utilised for extracting the correspondence addresses, the email addresses.

3. Database optimization

Databases are ubiquitous in the modern world. The notion of 'informational library', which is persistent, redundant and well distributed, has become the most important concept in the IT field. As a matter of fact, many people interact with a

database management system at a certain level, often without using a computer in every moment of the day.
On each access undergo millions of data transfers, database optimization being a key researching domain for university research institutions, as well as for corporate organizations. From a software development company point of view, the relational databases often serve the software applications in that domain, and the lack of optimization sustains significant costs for both the customer and the company. With millions of data transfers per second, the optimization comes as a surprise and thus represents a key research domain.

The database optimization allows a better configuration and faster searches results. Occasionally the database may present problems such as the failing to provide the requested result, or slow execution. That may make the acquisition of a server necessary. A similar role might have the operating system under which the database cannot be optimized.

The current database infrastructure can be revised thus establishing the best optimization approach and planning for better working environment efficiency.

Through the execution of a database quality control, it can be optimized without duplicates and with high integrity.

Fig. 1. Database schema

Nowadays, this optimization represents a real challenge, especially when the software is constantly changing. However, the database administrators offer relevant solutions to meet their clients' requirements.

Database administration applications

There are different approaches to database administration, and there also are different ways of optimizing the databases for performance boost, which will also improve the used server. The optimization will depend on the database management system. Each system has its own facilities for optimization. There are programs which have the role of collecting and analysing the required data in order to use the optimization process. These applications will be used in a more alert way as the optimization of the database will become increasingly noticeable. The database systems become more and more important so a continuous database update is mandatory in order to keep up with the changes in the IT domain.

Indexing

One of the ways in which a database can be optimized is indexing. This is made to increase the performance of the queries, which can vary from a database to another, but, generally speaking, all of them benefit from efficient indexes. The efficient indexes allow the queries to avoid scanning the entire structure tables in order to identify the solution. This can be realised with the Microsoft SQL servers.
The SQL server has been made for such sets of indexes. Moreover, its update is permanent, in order to allow the most efficient decisions in query processing. The expert can provide suggestions regarding the way in which the performances of the queries can be increased. The performance of the database must also be updated, so the changes in the dynamic systems have to be taken into account.

A database management system such as Oracle offers its own way of updating. It includes an SQL-type 'adviser' and additionally an access 'adviser'. Those are used to improve the SQL, which is used in package applications. It uses samples in order to collect the necessary data for updates.

The optimization is one of the important ways in which you can keep your systems at optimal performances. They can have different names, but essentially they contribute to the performance increase of the system.

The database optimizers are included in the software which the web holders can use. They represent more complex ways that only IT specialists may use. Nowadays, the applications offer characteristics that increase the optimization's efficiency. In order to be able to maintain the life cycle of the database, the holders have to assure that their databases are advanced.

Using indexes in a database for optimization

A database index is a physical access structure for a database table which works as its name suggests: it is a sorted file that informs the database of the whereabouts of the registrations, which are located on the disc. In order to better understand what an index does, please consider reading a textbook. In order to find a certain section, the reader can read the book until he identifies what he is looking for, or alternatively can check the 'contents' and find the desired section. A database index can work much longer than a textbook index. Adding adequate indexes for large tables is the most important part in optimizing a database. The creation of a unique index for a large table which contains no indexes can reduce the execution time of a query considerably.

*As example, we shall take the following scenario: supposing we have a table of a database named '**EMPLOYEES**' with 100.000 registrations. If we want it to execute the next simple query on this un-indexed table:*

```
SELECT First Name, Last Name FROM
EMPLOYEES WHERE EmpID=12345;
```

For the purpose of identifying the registration of the employee with the ID aforementioned, the database has to scan the entire 100.000 registrations in order to return the correct result. This way of scanning is usually known as a full-scan of the table. Fortunately, a database developer can create an index on the EmpID column to prevent such scans. Furthermore, in the case of a unique constraint of this domain, the database will compile the physical address of each employee in a table. As such, the scanning becomes pointless, and the localization of the registration is made in real time. After the developer adds this index, the database can locate the registration of the employee with EmpID=12345, which is a potential reduction of 100.000 operations.

Types of indexes

Indexes fall in one of the two categories: clustered or nonclustered. The main difference between the two categories is that the nonclustered indexes do not affect the indexes' ordering located on the hard while the clustered indexes do. Because clustered indexes do affect physically the order of the registrations from the disc, there can be an indexed cluster for each table. The same restriction cannot apply to nonclustered indexes, thus creating space on the disc is possible (though it does not represent the best solution).

Cost estimating for optimized databases

The cost estimating is the process of applying a consistent and significant execution measure of the costs for a certain query. Different metrics can be used for this purpose, but the most relevant

and the most common metric is **the number of block accesses query carts.** Since on the disk the inputs/outputs represent a time-consuming operation; therefore the objective is to minimize the number of block accesses, without the sacrifice of functionality.

Estimation of Select operations cost, estimation of Join operations cost, nested Loop, single Loop (using an index) and sort-merge Join can be taken into account.

There are a series of database optimizing methods. Each has ultimately as its result the reduction of the algorithm's complexity. One of the techniques used for that is the Greedy techniques.

The ***greedy*** algorithms are generally simple and are used in optimization issues (for example – finding the easiest path in a graph). In most situations we have:

- Lots of elements (vertices of the graph, works in progress etc.);
- A function that checks whether a mass of candidates is a possible, not necessarily optimal, solution;
- A function that checks whether is possible to complete it for a mass of candidates in order to a possible, not necessarily optimal, solution;
- A selection function that chooses the best unused element at any given time;
- A function that notifies the user that a solution has been reached.

To solve the problem, a *greedy* algorithm builds the solution step-by-step.

The Greedy technique states that the number of configurations (of the nodes and arcs in the graph) is exponential with the number of the candidate structures of the workload; at the same time the main algorithm is not feasible in the case of workloads with a large number of candidate structures.

At the roots of the GREEDY technique of number of configuration reductions stands the GREEDY algorithm:

- If the workload is a sequence then the algorithm is named GREEDY-SQ;
- The GREEDY-SQL algorithm uses a UnionPar function;

The UnionPar function

- Allows 2 solutions p_1= [a_1, S_1,..., a_N, S_N, a_{N+1}] și p_2= [b_1, S_1,..., b_N, S_N, b_{N+1}]
  For the sequence [S_1,...,S_N];
- Generates a new solution for the same sequence;
- At each K stage, there are generated additional configurations, starting from a_k and b_k configurations which are added in the graph;
- The exit is the shortest path in the generated graph.

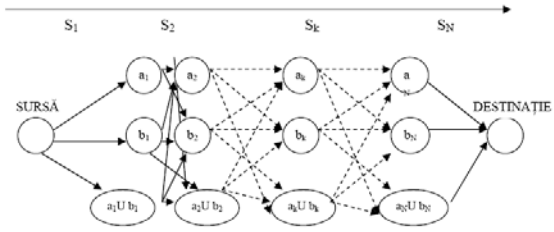

Fig. 2. Graph generated by the function UnioPair

The GREEDY-SEQ algorithm

Step 1. For each structure from S= {s_1, s_2,..., s_M} the optimal solution is found using the main algorithm. There are a lot of P solutions for individual structures.
Let P={p_1,..., p_M} and p_i= [a_{i1}, S_1,..., S_N, a_{iN+1}];
Step 2. Let C be the amount of all the configurations over the individual structures;
Step 3. On the P amount the greedy search is run.

Step 3a. Let r = [c_1, S_1,...,c_N, S_N, c_{N+1}] solution from P where **COST(r)** minimum. P=P-{r}.

Step 3b. We choose **s from P** for which **t=UnionPar(r,s)** has the cost of execution minimal for all elements from P, and **COST(t) < COST(r).**

If s does not exist proceed to **Step 4.**

P=P-{s}, P=PU{t} **goto Step3a.**

Step 4. The graph with all the configurations from P from that stage will be generated, after which the algorithm for the minimum path is run and the solution is given.

4. Conclusions

We use algorithms and techniques which lower the complexity of the centralised databases. This document has as purpose a better perception of the database optimizations for the developer, as well as the way in which a database (ex. DBMS) formulates executional strategies for different types of queries, even though the presented examples are limited in scope. It should also be noted that a well-created database should contain indexes and criteria for the selection of the columns for indexes.

References

[1] I. Lungu and A. Bara, *Executive Information Systems*, ASE Printing House, Bucharest, 2007.

[2] I. Lungu and I. Tanase, *Optimizing queries in relational databases*, Journal Informatica Economica, nr. 1(13)/2000.

[3] T. Marston, *The Relational Data Model, Normalisation and effective Database Design*, 2005.

[4] J.Date. *An introduction to Database Systems*, Addison Wesley, 2004.

[5] http://www.databaseguides.com/

[6] http://www.vinrcorp.com

Adrian Ghencea is Lecturer at Titu Maiorescu University, Faculty of Economics. He has a degree in Finance and Banking, master of ASE Bucharest – International Accounting Masters, master of University Bucharest – Development and Implementation of Web Services Masters. He is currently a Ph. D. candidate in Computer Science at ASE Bucharest and lecturer on computer science disciplines. Teaching courses such as Economic Informatics, Programming whit Visual Basic.net, Databases. He wrote scientific papers as well: Architecture of Computer Systems at the Level of Economic Agents – Trends. Mathematical Model for Modularisation of Applications Based on Object - oriented Design Metrics; Cyber-warfare and Business Development; Web 2.0 and Business Promotion; Adaptive Information Systems - the premise for Self Learning Systems etc.

Immo Gieger is a professor at Bodenstedt-Wilhelmschule Peine, Germany. Teacher and Assistant Director of this institution he became noted for various scientific and educational activities. Teach a wide range of disciplines including the computer science