THE BUCHAREST UNIVERSITY OF ECONOMIC STUDIES

DATABASE SYSTEMS JOURNAL

Vol. V, Issue 2/2014

LISTED IN

RePEc, EBSCO, DOAJ, Open J-Gate,
Cabell's Directories of Publishing Opportunities,
Index Copernicus, Google Scholar,
Directory of Science, Cite Factor,
Electronic Journals Library

BUSINESS INTELLIGENCE

ERP

DATA MINING

DATA WAREHOUSE

DATABASE

ISSN: 2069 – 3230
dbjournal.ro

Contact
Calea Dorobanţilor, no. 15-17, room 2017, Bucharest, Romania
Web: http://dbjournal.ro
E-mail: editordbjournal@gmail.com; editor@dbjournal.ro

CONTENTS

Integrating BI Tools in an Enterprise Portal for a better Enterprise Management

Gianina MIHAI
„Dunarea de Jos" University, Galati, Romania
grizescu@yahoo.com

Access to information has always been a stringent necessity for any organization. This necessity has become more stringent in the actual economic context, as decisions must be taken in the shortest time. That is why, since the early sixties, Decision Support Systems emerged. They have evolved together with the client-server technology, during the eighties, towards today's Executive Support Systems. Nowadays, these systems tend to be replaced by Dashboards and Balanced Scorecards as individual BI tools or as integrated tools, which are part of more complex BI solutions, which fulfill the more and more sophisticated and growing information requirements from decision factors. In this paper, we shall present a partial integration solution of the BI tools in an enterprise portal created using Microsoft SharePoint Server technologies. Initially, the enterprise portal which we'll present in this paper was meant to be more a general solution for the informational integration of an organization and less than a portal with strong BI features. But, as we shall demonstrate by the means of this paper, the portal prototype, as it is now, can be rapidly expanded to an enterprise portal with full BI features.

Keywords: *Business Intelligence, Enterprise portal, Dashboard, KPI, Integrated reporting*

1 Literature review

In specialized literature, there are a lot of synonyms for the enterprise portal term: corporate portal, corporate information portal, business portal or enterprise information portal [1]. In [2], beside a part of the already mentioned synonyms, there are others, like: employee portal, enterprise intranet portal, business-to-employee portal. In addition to these, Kim et al. [3] adds other terms, like data portal or collaborative portal.

Regardless of the chosen term, the broad definition signifies *"a portal is a gate towards information, services, products etc. provide, with the help of the Internet, of the Intranet of an organization or of any other proprietary Intranet. A portal is a gate. It offers an access point towards a very broad area of resources and services, like: e-mail, forums, search engines or online shops"* [4]. The strategic goal of an enterprise portal is to "*provide a simple, usable interface to all who need to interact with the organization: employees, customers, providers and business partners who need to access content, to use applications and to collaborate with each other*"[5].

In the early 2000s, Gartner Group forecasted that by the end of 2001 more than half of all the important companies will use a portal as their main organizational and informational tool [6]. These assessments were also sustained by the numerous research studies which emphasized the importance of enterprise portals, considering them to be *"the most important business information management project for the next 10 years"* [7] or considered that it would become *"the next generation of software for desktop applications"* and *"its impact on information management will be as important as the railroads were for the industrial revolution"* [8]. It was natural to be so for the novelty, not so of the portals themselves, but especially for the novelty of the web technologies that barely had emerged. Nowadays, the interest in portal

technologies has dropped, which is partly due to the fact that web interfaces and web technologies have become very common, even for software applications which are used at transactional level in organizations.
Specialized literature speaks very highly of portals, from the point of view of the numerous advantages they provide to enterprises. Beside the fact that an enterprise portal can provide advanced search functions for the users, by the means of integrating different information sources and of the easier access to the enterprise software applications, it can also provide employees with tools which can help them find information and sometimes even knowledge they need to complete their tasks [9], [10]. Unlike other systems, enterprise portals provide information taking into consideration the roles of the users within the organization [11], thus diminishing the „informational overload", cutting down costs, improving their innovation capacities and providing Business Intelligence features [12]. A portal can also provide the following features: rationalizing business processes, increasing efficiency and productivity, and, thanks to the easy access to applications and relevant information, improving employee satisfaction. All this is possible thanks to the improved communication and collaboration between employees and workgroups across an organization [2], [6], [13], [14]. A portal can improve as well the external collaboration process with different business partners [6], [14].
Yet, all these advantages are still theoretical, as they weren't proven by any studies on real implementations, thus confirming or disproving them [14]. Although it's been a while since Dias has published his article, other empirical studies on these matters are very scarce. There are only the Detlor [9] and Detlor and Finn [15] studies [16].
Based on the experience gained while working on the portal referenced by this article, we consider that all the advantages above mentioned can be obtained if all the portal features are fully employed by all users.

2. Introduction

In tight connection with the BI field of activity, a more familiar term is "business intelligence portal". However, these portals are much more restrictive than enterprise portals, because they provide only one unique, secure web interface, for integrated, personalized BI applications, like: reports, data hypercubes, dashboards, scorecards etc. Beside BI specific applications, some portals also provide unstructured content integration instruments, or collaboration tools. Because of these limitations, most times BI tools are integrated into one single enterprise portal, especially if several other BI applications or portals are already integrated. All these can be contained into one single enterprise portal which can provide, through a single access point, personalized interfaces for each particular user.
Nowadays, thanks to the development of BI technologies, there are several ways to use BI tools in business processes among which we mention [17]:

1. Integrating BI tools in the operational applications already implemented in the organization;
2. Integrating analytical applications and BI tools in an enterprise portal;
3. Using Web Services in order to dynamically integrate BI tools both with internal operational applications and external applications from different business partners, thus supporting the collaborative commerce between organizations;
4. Implementing on-demand, event-driven processing for alerting systems, for real-time recommendation systems and for automating systems for different activities.

In this article, we shall approach the integration of BI tools in portals as an alternative to creating dedicated BI systems. We favor this approach as we

consider it to have a series of advantages, among which there are the speed and reduced costs of providing personalized BI tools to a great number of users. In addition to that, implementing an enterprise portal which is capable of offering tools like dashboards, scorecards or collaboration tools is much more appropriate for managers, decision factors and advanced users who need to analyze, evaluate and collaborate before making a decision.

3. Portal architecture

The solution we chose for implementing BI tools in an organization consists of a component of the enterprise portal we built as an organizational, integrating and collaborating, general-purpose solution. In order to build the portal, we used Microsoft Office SharePoint Server (MOSS) technologies, for several reasons: the modularity and reusability of the component technologies, the extensibility, scalability, and the service-oriented architecture (SOA) and, last, but not least, the complete and extremely complex enterprise architecture.

The general architecture of the solution, obtained by using MOSS consists of three levels, each with component servers, which have one of the three following roles: Web Server, Application Server and Database Server.

In order to build the portal and all its components, it was necessary to install, configure and implement several software components:

- Operating system - Windows Server 2003 SP2 or later;
- Web Server - IIS 6.0 or later and ASP.NET 2.0 or later;
- Database Server - SQL Server 2005 or later;
- E-mail Server – POP3 and SMTP;
- SharePoint Server - Microsoft Office SharePoint Server (MOSS) 2007, including Forms Services for creating sites;
- Report Server – SQL Server Reporting Services 2005 with SP2, including Reporting Services Add-in for SharePoint 2007, in order to integrate it with the SharePoint Server;
- Excel Calculation Services, Excel Web Access and Excel Web Services – which extend the portal functions in the direction of accessing external data sources in Excel files and integrating them in the portal. These, together with the SQL Server and the SQL Reporting Services, are the necessary components for building and implementing dashboards and scorecards.

Figure 1 depicts the architecture of the portal hosted on a single physical server.

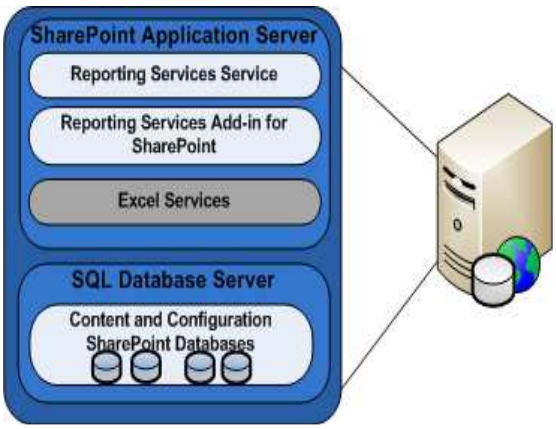


Fig. 1. Portal architecture

The main advantage of this solution is that the three-level logical architecture can be implemented on a great variety of physical architectures, from a single server up to a great number of servers, depending on the organization's dimension and specifics.

Our solution combines all of these elements, being able this way to provide organizational features which help solve critical issues: efficient management of the business content and processes, easy, real-time access to information, thus contributing to better decisions.

4. The intelligent reporting solution implemented in the portal

The reporting process must become an intelligent one. An intelligent reporting

system is one which turns data into information and the information into knowledge, supporting users into taking the best decisions. The reporting system which our solution provides by integrating SharePoint Server with Microsoft SQL Reporting Services is such an intelligent system, because it successfully addresses the following problems:

- *Integrating and consolidating different data sources* inside the enterprise and even outside the enterprise in order to be able to obtain relevant information and improve efficiency. This is done with the help of the integrated reporting solution;
- *Computing and reporting key performance indicators*. Integrating these indicators and presenting them in one single view makes them more suggestive and relevant in the decision-making process. For example, if one can see on one single display, in the form of a dashboard or of a scorecard, that the number of clients is dropping, while the number of employees is rising, the issue can be addressed more quickly;
- *Eliminating duplicate information.* Different applications in an organization mean that in fact the same data is stored in several places. Using our integrated solution, the organization can drastically decrease or even completely eliminate duplicate information.

So, the reporting functions of the portal address these problems and helps improve efficiency and the same time, provide better information features for the employees and decision factors.

SQL Server 2005 Reporting Services is a server reporting platform, which can be used to create and manage table-type reports, matrixes, diagrams or any other type, using data both from relational and multidimensional databases. If the Reporting Services server is integrated with the SharePoint server, the reports properties and elements are stored into the SharePoint content databases, providing a better integration of the two server technologies. This integration determines the way the content is stored, secured and accessed.

Figure 2 illustrates how the two servers, the SharePoint Server and the Report Server, are integrated and are working together during a report generating process.

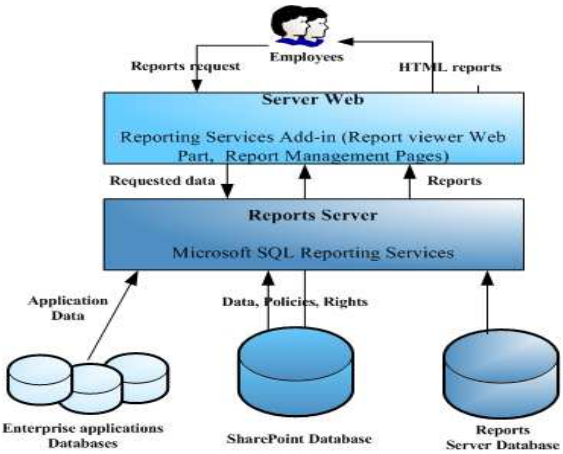


Fig. 2. The report generating process within the portal

When a report is opened in the portal, the SharePoint server connects to the reporting server, creates a session, prepares the report for processing, takes over the data, configures the report according to the display properties and displays it in the portal, in a Web part which is called a Report Viewer. As long as the report is opened, it can be exported into different other formats, other data detail levels can be requested, filters can be added, or another report can be accessed by the means of some connections included into the opened report. The interaction with the report, like exporting, for example is also managed by the reports server.

We also chose to implement a Reporting Center into the portal, as a solution for the integrated access to application data and for implementing dashboards and scorecards. A Reporting Center provides access to all users to the organization's data, by the means of reports, dashboards,

key performance indicators (KPIs) and data source libraries. The Reporting Center can also filter data before displaying it to the users. This allows customizing it, before displaying it to the users. For example, accountants can only see reports, key performance indicators and dashboards which deal with the financial and accountancy data, managers can only see synthesis information and reports etc. in the portal, data is filtered at the page-level and this is done by the means of Web part connections. The portal allows the employees to choose different data filtering conditions, like the calendar date, suppliers, clients or a certain category of reports.

The reports published in the reporting center are organized in report libraries. These reports can be created using data taken from any organizational application that supports an ODBC or OLEDB connection. The image in figure 3 presents a report library from the Reporting Center where there are published reports created using data from three different transactional applications:

- Saga – accounting application, Visual FoxPro database;
- ContabSQL – management application, Paradox database;
- GestBal – staff tracking and payroll application, MySQL database.

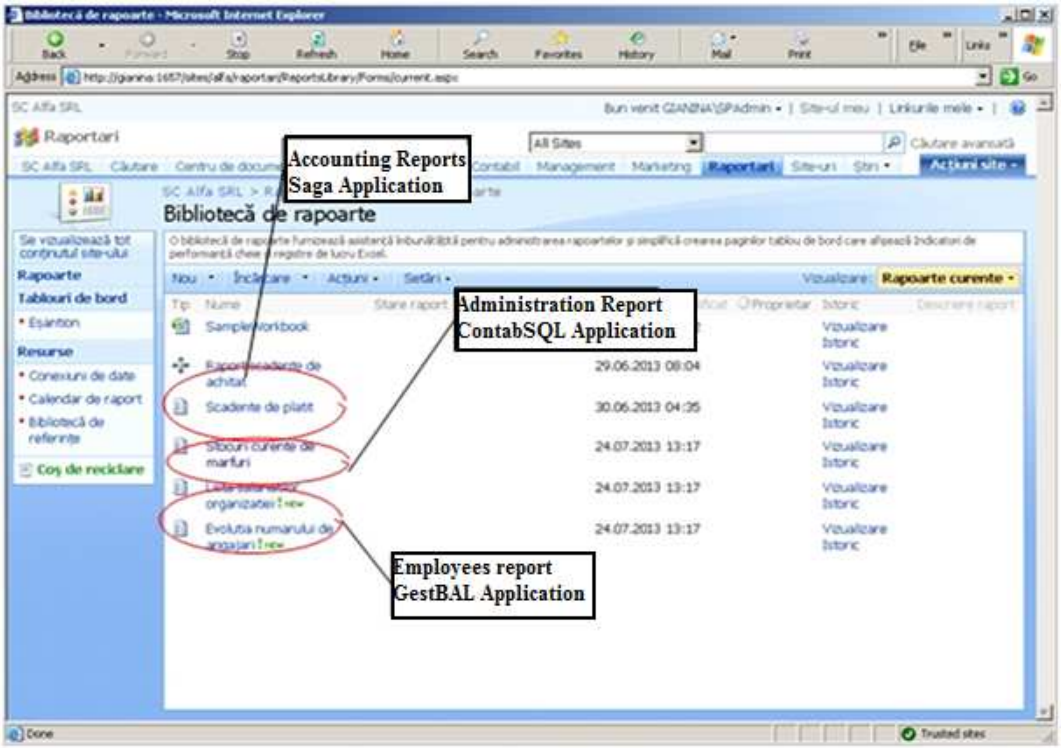
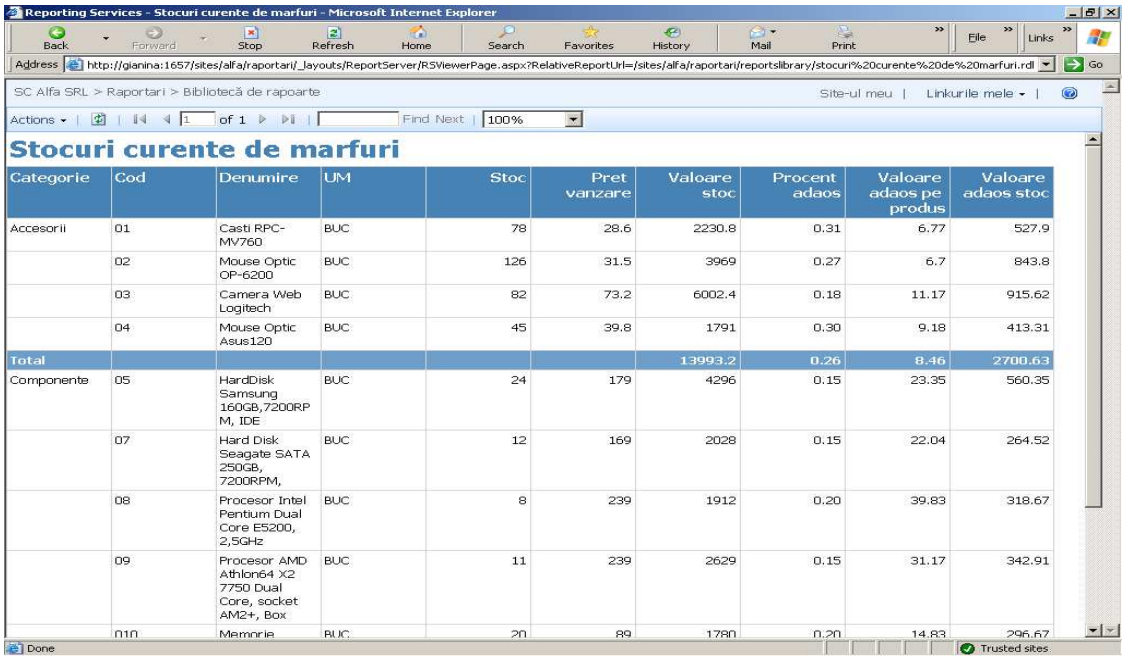


Fig. 3. Integrating reports from different applications into the portal

In spite of having different applications and data sources, the SQL Server Reporting Services can create reports from all of them and publish them in the portal, in only one library. Thus, employees are provided with access from a single point to data from all applications used in the organization. Once published in the reports library, the access to these reports is done in a controlled and secure manner. They can be viewed only by users with proper permissions, directly in the web-browser, by simply clicking on them. They can also be exported into other formats (pdf, xls, html, tiff, csv) in order to be used and processed offline (outside the portal). The image in figure 4 shows the current inventories report generated using data from the ContabSQL management

application and displayed in html format in the portal.



Categorie	Cod	Denumire	UM	Stoc	Pret vanzare	Valoare stoc	Procent adaos	Valoare adaos pe produs	Valoare adaos stoc
Accesorii	01	Casti RPC-MV760	BUC	78	28.6	2230.8	0.31	6.77	527.9
	02	Mouse Optic OP-6200	BUC	126	31.5	3969	0.27	6.7	843.8
	03	Camera Web Logitech	BUC	82	73.2	6002.4	0.18	11.17	915.62
	04	Mouse Optic Asus120	BUC	45	39.8	1791	0.30	9.18	413.31
Total						13993.2	0.26	8.46	2700.63
Componente	05	HardDisk Samsung 160GB,7200RPM, IDE	BUC	24	179	4296	0.15	23.35	560.35
	07	Hard Disk Seagate SATA 250GB, 7200RPM,	BUC	12	169	2028	0.15	22.04	264.52
	08	Procesor Intel Pentium Dual Core E5200, 2,5GHz	BUC	8	239	1912	0.20	39.83	318.67
	09	Procesor AMD Athlon64 X2 7750 Dual Core, socket AM2+, Box	BUC	11	239	2629	0.15	31.17	342.91
	010	Memorie	BUC	20	89	1780	0.20	14.83	296.67

Fig. 4. Report of current inventories

Any employee with proper permissions can see the inventory at any moment, even without access to the management application and, most of all, without having to wait for a report from the Sales Department.

Connecting to external data sources in Excel files or in databases is a necessity for an organization, as it needs to cover the lack of some functions in the already implemented applications. Together with the report, a connection to the database is also published in the portal, into the data connection library. These data connection libraries can be used to manage connections created both with Microsoft Office applications (.odc – office data connection files) and SQL Reporting Services (.rds – report data source files). Once saved in the data connection libraries, other users can use these connections in order to create new .xls files or new reports, without having to possess the necessary knowledge required to connect to external databases. Also, by the means of libraries, access to these data sources can be monitored and controlled.

The image in figure 5 shows a dashboard built with the help of the integrated reporting solution, which gives decision factors the possibility to watch in real-time the organization's capacity to deal with overdue payments during a 10 days period. The dashboard has four components, all linked to the accounting application database through an ODBC connection.

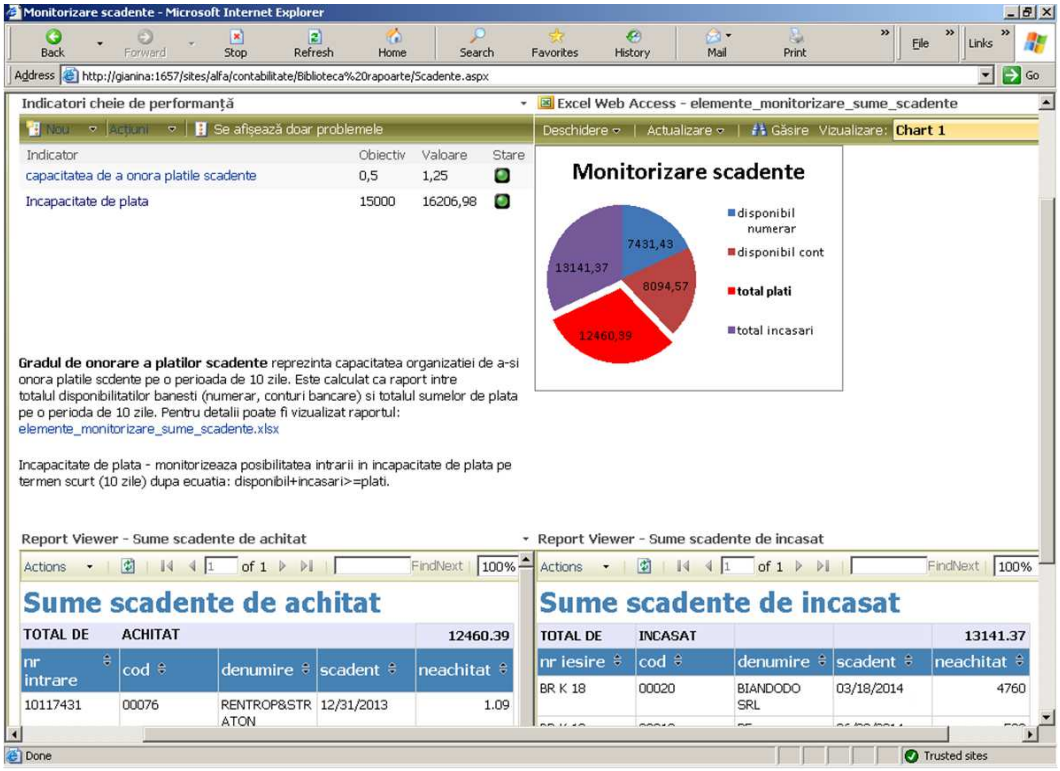


Fig. 5. Overdue payments monitoring dashboard

The four components of the dashboard are:

1. *Key performance indicators*. The dashboard computes and monitors two performance indicators:
   - *The capacity to deal with overdue payments*, calculated using the formula:

$$\frac{(Available\ Cash\ +\ Available\ Bank)}{Total\ Amount\ to\ be\ Paid}$$

     The indicator is considered to be good if the available cash can cover 50% of the total amount to be paid for the next 10 days. If the available cash drops to 30% from the total sum to be paid, the user will be notified by a yellow indicator (in the form of a traffic light).
   - *Insolvency*. This indicator shows the organization is on the edge of insolvency, by adding to the available cash, the collections for the next period. The indicator is monitored using the inequality:

$$Available\ Cash\ +\ Available\ Bank\ +\ Collections\ -\ Payments \geq nr$$

     Analyzing cash collections and payments during a 10 day period, there can be established an optimal level for this indicator.
2. *Overdue payments monitoring diagram* – shows in a graphical manner the total available cash in the cash register, in the bank account, the collections and the payments for the next 10 days.
3. *Reports* created using data took directly from the accounting application, which provide details on the collections' origin and the payments' beneficiaries for the next 10 days.

The *Overdue payments monitoring dashboard*, by its components, allows decision factors to watch collections and payments continuously, so that they can avoid penalties for the overdue payments or even the insolvency of the organization. Such a dashboard can improve the efficiency and acuity of the decision factors, providing them access to relevant and accurate data they need to successfully complete their tasks. The most important thing is that any of the dashboard's components can be connected to the

database of any of the organizational applications by the means of an ODBC or OLEDB connection. This way, dashboards can integrate data from several applications, with decision factors being able to get exact, meaningful, real-time information for their activities, regardless of the application the data comes from.

5. Limits and further development

According to the MicroStrategy company, one of the biggest BI solution producers and integrators in the IT field, a complete BI solution should provide a set of instruments that are capable of offering reports and also monitoring and analyzing features using data from the organizational data storage to a very broad category of users, both internal and external. In order to be able to do that, the portal should integrate five categories of BI tools [18]:

1. *Dashboards and scorecards* – these BI tools cover all monitoring requirements for an organization, from management and decision factors level, down to simple employees and even providers (external users);
2. *Reporting tools* – used to create operational and financial reports in standard formats (.pdf, .xls, .txt etc.) and in Web format and distribute them to both users and decision factors in the organization and external users;
3. *Multidimensional data analysis tools* – these tools are good for analysis which can be forecasted. They include mainly OLAP cubes, which allow ad-hoc data interrogation on multiple dimensions and different data subsets, obtained by specific slice and dice operations and also by data drill down from the highest aggregation level to the lowest detail level (transactional level);
4. *Predictive data analysis and advanced data analysis tools* – they can discover unexpected data behavior and can make forecasts, using set theory techniques and mathematic and statistic functions. These tools, also known as data mining tools, can perform advanced analysis, predictive analysis, hypothesis and forecast testing;
5. *Proactive warning and notification tools* – they are used for automated distributing of great number of reports, scorecards or warnings to a great number of users, in a pro-active, centralized manner, according to a previously established schedule; they can also send predefined or on-demand triggers based on a subscription system.

As one can see, the prototype portal we built has only reporting tools, dashboards and scorecards implemented and also a few warning and automation features for distributing reports throughout an organization. In order to implement the other BI tools, multi-dimensional analysis and data mining tools, we shall have to install and integrate the Microsoft Analysis Services component in SharePoint Mode; we shall also have to install and configure the Excel Services Application Data Model in order to integrate it with the Analysis Services Server and integrate it in the portal. The future, three level architecture of the portal will look like the image in figure 6.

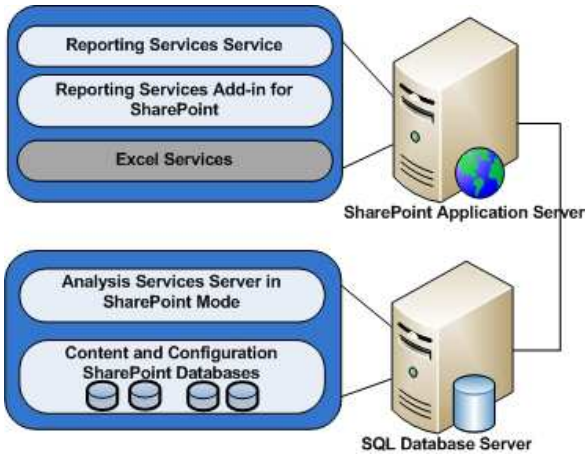


Fig. 6. The future three-level architecture of the portal

By adding an analysis server, but especially for building practical solutions which can imply even creating a data

warehouse in the organization in order to be able to completely benefit from the power of OLAP and data mining tools, the one single server architecture (see figure 1), will not be appropriate anymore, even for small organizations.

6. Conclusions

While organizations are more and more dependent on information, collaboration and integrating multiple, both internal and external, data sources, decision factors must face a growing challenge: to manage information provided by lots of carriers: events, digits, documents, reports etc. In a more and more dynamic working environment, our solution provides the necessary tools it takes to manage, organize, share and provide efficiently all this content to both decision factors and simple users in an organization.

Created using SharePoint technologies, the solution we presented in this paper provides an intelligent organizational reporting process and also creating dynamic, interactive dashboards, which can combine and integrate data from multiple sources and can provide meaningful information to the users, taking into account their roles in the organization and, consequently, their particular informational needs. By combining Web Parts, filters, key performance indicators (KPIs), diagrams, reports and other elements, these dashboards are capable of offering ideal combinations of information, formatted and presented according to the particular requirements of each user.

Extending the portal's functions by implementing the Microsoft Analysis Services in SharePoint Mode component will lead to obtaining a complete BI solution integrated into the enterprise portal.

References

[1] C. Finkelstein and P. Aiken, *Building Corporate Portals with XML*, USA, McGraw-Hill, 2000, p.16;

[2] H. Benbya et al., „Corporate Portal: A Tool for Knowledge Management Synchronization", *International Journal of Information Management*, Vol 24, pp. 201-220, 2004;

[3] Y.J. Kim et al., A Knowledge Management Perspective to Evaluation of Enterprise Portals, *Knowledge and Process Management*, Vol 9, No 2, pp.57-71, 2002;

[4] T. Korhonen and A. Ainamo, *Handbook of product and service development in communications and information technology*, Boston, Ed. Kluwer/Springer, 2003;

[5] W. Wojtkowski, „Collaborative Enterprise Portals", *Encyclopedia of Portal Technologies and Applications*, USA, Ed. Information Science Reference, 2007;

[6] B. Detlor, "The Corporate Portal as an Information Infrastructure: Towards a Framework for Portal Design", *Int. Journal of Information Management*, Vol 20, No 2, pp.91-101, 2000;

[7] H. Collins, *Corporate Portals: Revolutionizing Information Access to Increase Productivity and Drive the Bottom Line*, New York , AMACOM, 2001;

[8] T.M. Koulopoulos (1999, April), "Corporate Portals: Make Knowledge Accessible To All", *Information Week Online*, Available: http://www.informationweek.com/731/31erall.htm;

[9] B. Detlor, *Towards Knowledge Portals: From Human Issues to Intelligent Agents*, Dordrecht, The Netherlands, Kluwer Academic Publisher, 2004;

[10] C. Terra and C. Gordon, *Realizing the Promise of Corporate Portals*,

New-York, Ed. Butterworth-Heinemann, 2003;
[11] D. Ben-Arieh and M. Pollatscheck, "Analysis of Information Flow in Hierarchical Organizations", *International Journal of Production Research*, Vol. 40, No. 15, pp.3561-3573, 2002;
[12] L.F. Sugianto and D. Tojib, "Modeling user satisfaction with an employee portal", *International Journal of Business*, Vol. 24, pp. 339–348, 2006;
[13] M. Rahim and M. Singh, (2006), "Understanding Benefits and Impediments of B2E E-Business Systems Adoption: Experience of Two Large Australian Universities" in *IADIS International Conference e-Society*, Dublin, Ireland, 2006;
[14] C. Dias, "Corporate Portals: A Literature Review of a New Concept in Information Management", *International Journal of Information Management*, Vol. 21, No. 4, p.269-287, 2001;
[15] B. Detlor and K. Finn, "Towards a framework for government portal design: The government, citizen, and portal perspectives" in *Electronic government: Design, applications, and management*, Hershey, Ed. Idea Group, pp. 99-119, 2002;
[16] F. Landqvist and D. Stenmark, "Portal Information Integration and Ownership misfits: A Case Study in a Tourism Setting", *39th International Conference on System Sciences*, Hawaii, published in IEEE Computer Society, 2006;
[17] M. Ferguson (2006, July), "*Techniques for Integrating Business Intelligence into the Enterprise*", published in *www.businessintelligence.com.* , Available: http://www.b-eye-network.co.uk/view/3110;
[18] *Enterprise Business Intelligence Improving. Corporate Performance Through Integrated Reporting, Analysis, and Monitoring*, MicroStrategy white paper, USA, Virginia, 2005. Available: http://www2.microstrategy.com/download/files/whitepapers/open/MicroStrategy_EnterpriseBusinessIntelligenceImproving.pdf.

Gianina MIHAI is a graduate of the "Dunarea de Jos" University in Galati, Economics and Business Administration Faculty. She specializes in Economic Informatics, which she graduated in 2000. Beginning with July 2010 she holds a Ph.D. in Economics, Accountancy to be more specific, specialized in Management Information Systems.
She is also the author or co-author of eight books, university courses, practical studies and over 25 papers and articles on economic informatics, accounting information systems and integrated information systems, published in specialized journals or presented during national and international conferences.
Nowadays she is a lecturer at the "Dunarea de Jos" University in Galati, Romania, Faculty of Economics and Business Administration, Business Administration Department.
Among her scientific fields of interest there are: databases, information systems integration technologies, collaborative technologies, ERP systems and accounting information systems.

SGA Dynamic Parameters: The Core Components of Automated Database Tuning

Hitesh KUMAR SHARMA, Aditya SHASTRI, Ranjit BISWAS, Sanjeev KUMAR SINGH
[1]University of Petroleum & Energy Studies
[2]Banasthali University
[3]Hamdard University
[4]Galgotia University Noida
hkshitesh@gmail.com, adityashastri@yahoo.com, ranjitbiswas@yahoo.com, sksingh8@gmail.com

The efficient use of primary memory is one of the major key for the good performance achieved from the any kind of server. The management of the various components stored in the main memory is the key challenge to get the desired throughput from an application running on the server. As we know the DBMS mostly works on client-server architecture. So the memory management for the DBMS's components stored in the main memory of the server is the critical task for a DBA. The DBA should have the knowledge of the components those are stored in main memory during runtime. This paper helps the DBA to get the detailed description of these core components of DBMS.
Keywords: *SGA, DBMS, DBA, Dynamic Parameters.*

1 Introduction

The Complexity of the database is increasing day by day. Due to the Exponential growth of the amount of data and its complexity, the responsibilities of DBA has been increased in the same ratio.
One of the major responsibilities of DBA is to make the database available 24*7 for the user and the response time should be minimum. There are many components in a DBMS those are responsible for poor response time. These may be categories as software component (database design, SQL query parsing and optimize etc.) and hardware component (disk, main memory, network component etc.). Out of these many software and hardware components the major part played in performance tuning is the memory management component.
In this paper we will explain in detail the role of memory management in performance enhancement of a DBMS. The System Global Area (SGA) management impacts a lot on the performance of a DBMS. It is a big challenge in itself to predict and allot a right amount of memory for different components of SGA. The memory captured by an SGA is called as Instance of the database server. The management of this instance is done by several dynamic SGA Parameters or instance parameters. This paper will give the detail overview of SGA and its dynamic parameters.

2. System Global Area (SGA)

Inside the Instance of a database server, data is stored in two places: in memory and on disk. Memory has the best performance but also has the highest cost. Disk, on the other hand, can store vast amounts of data and cost effective but has very slow performance relative to memory. Due to the better performance, it is desirable to use memory to access data whenever possible. But because of the vast amounts of data usually accessed and the number of users who need this data, there is a lot of contention on this resource. To make most effective use of memory, you must achieve a balance between the memory used for DBMS caching and the memory needed by the users. The **System Global Area** (SGA) is the commonly shared main memory space on database server that is globally

shared by the clients connected to the server. If multiple users are connected to the database servers at a time than the pool of memory allotted to the server is called SGA and the SGA with background processes is called an Instance. An SGA and Oracle processes constitute an Oracle instance. Oracle automatically allocates memory for an SGA when you start an instance, and the operating system reclaims the memory when you shut down the instance. Each instance has its own SGA. The SGA is read/write. All users connected to a multiple-process database instance can read information contained within the instance's SGA, and several processes write to the SGA during execution of Oracle.

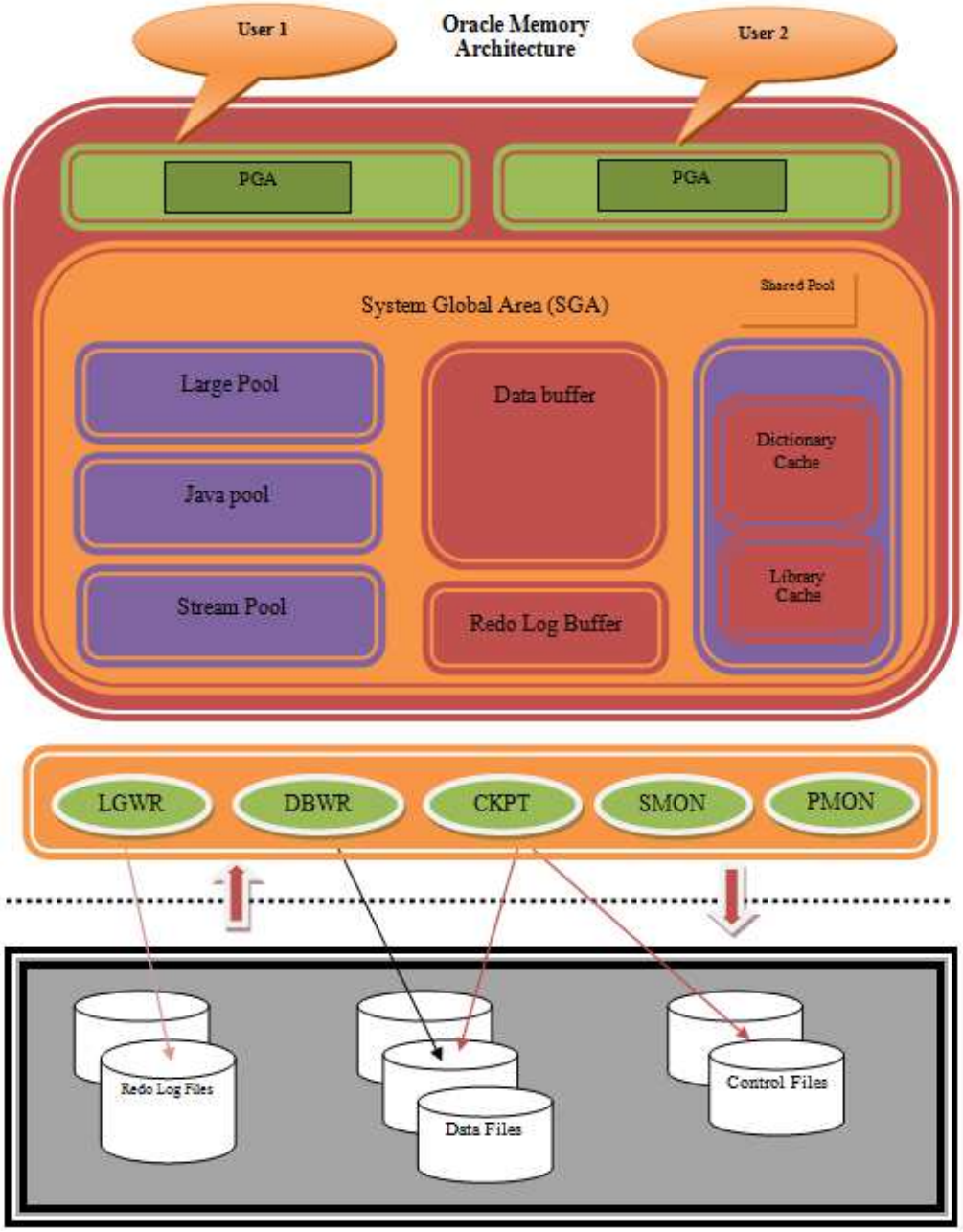


Fig.1 Oracle Memory Architecture

The SGA contains the following data structures:

- Database buffer cache;
- Redo log buffer;

- Shared pool;
- Java pool;
- Large pool (optional);
- Streams pool;
- Other miscellaneous information.

The Database Server Instance (DSI) also reserved some fixed space that is used to store instance information and the information of background processes running behind. User data is not stored here. The SGA includes information communicated between processes, such as locking information. If the system uses shared server architecture, then the request and response queues and some contents of the PGA are in the SGA.

The detailed description of the various components of SGA is given below.

2.1. Database Buffer Cache

This component of SGA holds the data blocks fetched from the disk to satisfy user's query. If the DML operations are performed on data blocks then The DML operations on data objects are first performed on these blocks and transferred to the data files by the DB Writer (i.e. DBWR) processes.

Free buffer, pinned buffer and dirty buffer are the part of LRU list. Free buffer will take place towards the LRU end of the list and dirty buffer will take place towards the MRU end. The Diagram (Figure 2) shows the detailed view of Data Buffer Cache. This diagram clearly shows the transferring the data from data files present in the disk to the various parts of data buffer cache. It also shows the role of LRU list and Write list. The transferring of the data processed by background process.

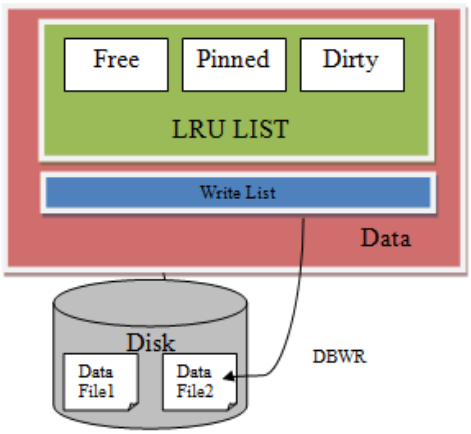


Fig.2 Data Buffer Cache (Components)

When a process requires data, it starts looking for it in the data buffer. If it finds that data, it is a cache hit otherwise it is cache miss. In the event of a cache miss, the process has to copy the data from data file into the LRU List. Before copying, the process will first start looking for free space in the LRU list starting from the LRU end. As the process starts to scans for free space, if it hits a dirty data, it copies that over to the write list. It continues until it has found enough free space or if it hits a threshold to search. If it hits the threshold, it asks DBWR process to write some of the data blocks from write list to data files and free up space. If it gets the required free space, it reads the data into the MRU end of LRU List. Whenever an Oracle process accesses any of the data in the LRU list (cache hit), the data is moved to the MRU end of that buffer. Over the time, the older data (except for full table scans) moves to the LRU end of the buffer.

2.2. Redo Log Buffer

Redo log buffer is used for recovery management. Whenever an SQL operation is performed on the database, a corresponding entry is made in redo log buffer. Due to the limited size of buffer the entries will be transferred to the online redo file present in disk using the LGWR

process. If some failure occurs due to abnormal termination then the data can be recovered using this buffer or online redo file.

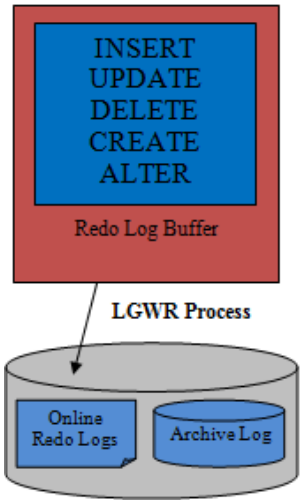


Fig.3 Data Buffer Cache (Block Movement)

The initialization parameter LOG_BUFFER determines the size (in bytes) of the redo log buffer. In general, larger values reduce log file I/O, particularly if transactions are long or numerous. The default setting is either 512 kilobytes (KB) or 128 KB times the setting of the CPU_COUNT parameter, whichever is greater.

2.3. Shared Pool

The shared pool component of SGA contains the parsed SQL statements and the metadata about the database objects. It divided into two major parts.

- Library Cache;
- Data Dictionary Cache.

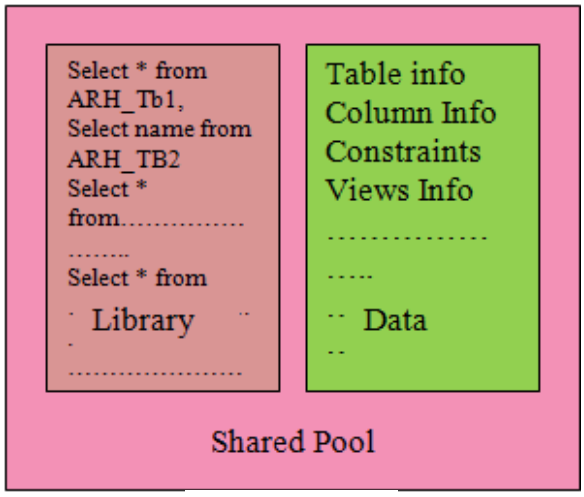


Fig.4 Shared Pool

The total size of the shared pool is determined by the initialization parameter SHARED_POOL_SIZE. The default value of this parameter is 8MB on 32-bit platforms and 64MB on 64-bit platforms. We can increase or decrease the value of this parameter as per the performance need.

2.3.1. Library Cache

The library cache stores the parsed SQL statements. If the SQL statements parsed succesfully then it gets entry in library

cache to reduce the parsing time in future for same SQL query. It is the reserved space in shared pool component of SGA. The diagram (Figure 4) shows the information stored in library cache. Some SQL queries has been shown in the diagram. The queries are stored in parsed form. The use of bind variable in SQL query make this space more useful Otherwise the same query will store multiple time just because of the difference in data used in where condition. Shared SQL areas are accessible to all users, so the library cache is contained in the shared pool within the SGA.

2.3.2. Dictionary Cache

The data dictionary is a collection of database tables and views containing reference information about the database, its structures, and its users. Oracle accesses the data dictionary frequently during SQL statement parsing. This access is essential to the continuing operation of Oracle.

The data dictionary is accessed so often by Oracle that two special locations in memory are designated to hold dictionary data. One area is called the **data dictionary cache**, also known as the **row cache** because it holds data as rows instead of buffers (which hold entire blocks of data). The other area in memory to hold dictionary data is the library cache. All Oracle user processes share these two caches for access to data dictionary information.

2.4. Large Pool

The database administrator can configure an optional memory area called the large pool to provide large memory allocations for:

- Session memory for the shared server and the Oracle XA interface (used where transactions interact with more than one database);
- I/O server processes;
- Oracle backup and restore operations.

By allocating session memory from the large pool for shared server, Oracle XA, or parallel query buffers, Oracle can use the shared pool primarily for caching shared SQL and avoid the performance overhead caused by shrinking the shared SQL cache. In addition, the memory for Oracle backup and restore operations, for I/O server processes, and for parallel buffers is allocated in buffers of a few hundred kilobytes. The large pool is better able to satisfy such large memory requests than the shared pool. The large pool does not have an LRU list. It is different from reserved space in the shared pool, which uses the same LRU list as other memory allocated from the shared pool.

2.5. Java Pool

Java pool memory is used in server memory for all session-specific Java code and data within the JVM. Java pool memory is used in different ways, depending on what mode the Oracle server is running in. The Java pool advisor statistics provide information about library cache memory used for Java and predict how changes in the size of the Java pool can affect the parse rate. The Java pool advisor is internally turned on when statistics_level is set to TYPICAL or higher. These statistics reset when the advisor is turned off.

2.6. Streams Pool

In a single database, you can specify that Streams memory be allocated from a new pool in the SGA called the Streams pool. To configure the Streams pool, specify the size of the pool in bytes using the STREAMS_POOL_SIZE initialization parameter. If the size of the Streams pool is greater than zero, then any SGA memory used by Streams is allocated from the Streams pool. If the size of the Streams pool is zero, then the memory used by Streams is allocated from the shared pool and may use up to 10% of the shared pool.

3. SGA Parameters

As we have discussed above that the SGA is the collection of various components.

The components have been divided on the basis of their requirements to achieve better functionality. The components have different memory requirement to store different kind of information. The accessibility mode, time and background process is also different for different component. By altering the memory assignment for them will have positive or negative impact on DBMS performance. Most of the DBMS provides the list of parameters associated with each component of SGA. By altering the value of these components the DBA can control the memory assignment for each component separately.

The following table lists the name of parameters with its associated SGA component.

Table 1. SGA Components and Corresponding SGA Parameters

SGA Component	Initialization Parameter
The buffer cache	DB_CACHE_SIZE
The buffer cache	DB_BLOCK_BUFFER
The buffer cache	DB_BLOCK_SIZE
The buffer cache	DB_KEEP_CACHE_SIZE
The buffer cache	DB_RECYCLE_CACHE_SIZE
The shared pool	SHARED_POOL_SIZE
The Redo Log Buffer	LOG_BUFFER
The large pool	LARGE_POOL_SIZE
The Java pool	JAVA_POOL_SIZE
The Streams pool	STREAMS_POOL_SIZE

The value of these parameters can be changed using **alter system** command. By altering the value of these parameters the DBA can increase or decrease the value of corresponding part of SGA.

4. Dynamic SGA Parameters

The parameters listed above can be changed by alter system command but all of them cannot give their changed effect on running instance. Some of them give their effect on restarting the instance. But the rest parameters can affect the instance in running state. These parameters who gave their effect on running instance are called Dynamic parameters. Since these parameters are used to alter the SGA components hence these are called Dynamic SGA parameters.

Table 2. SGA Components and Corresponding Dynamic SGA Parameters

SGA Component	Initialization Parameter
The buffer cache	DB_CACHE_SIZE
The shared pool	SHARED_POOL_SIZE
The Redo Log Buffer	LOG_BUFFER

SGA Component	Initialization Parameter
The large pool	LARGE_POOL_SIZE
The Java pool	JAVA_POOL_SIZE

These parameters are also called Auto Tuned parameters because these can be changed at runtime by using any auto executed application.

4.1. DB_CACHE_SIZE

This parameter is used for allocating the size of database buffer. By changing the value of this parameter the DBA can change the size of data buffer. The I/O overhead can be reduced by allocation sufficient memory to data buffer using this parameter. The value is calculated by multiplying No. of Blocks to Block size.

```
DB_CACHE_SIZE = 4M * Number of
CPU * Granule
```

The value larger than this is rounded up to the nearest granule size.

4.2. SHARED_POOL_SIZE

This parameter is used to assign the memory space to shared pool. The shared pool contains parsed SQL queries, PL/SQL procedures, Triggers, cursors, and other structures. If you set PARALLEL_AUTOMATIC_TUNING to false, then Oracle also allocates parallel execution message buffers from the shared pool. Larger values improve performance in multi-user systems. Smaller values use less memory.

4.3. LOG_BUFFER

LOG_BUFFER specifies the number of bytes allocated to the redo log buffer. Larger values reduce I/O to the redo log by writing fewer blocks of a larger size. Particularly in a heavily used system, this may help performance. In general, larger values for LOG_BUFFER reduce redo log file I/O, particularly if transactions are long or numerous. In a busy system, a value 65536 or higher is reasonable.

4.4. LARGE_POOL_SIZE

This parameter is used to assign the memory for backup and recovery of the database. Parallel execution allocates buffers out of the large pool only when PARALLEL_AUTOMATIC_TUNING is set to true. You can specify the value of this parameter using a number, optionally followed by K or M to specify kilobytes or megabytes, respectively. If you do not specify K or M, then the number is taken as bytes.

4.5. JAVA_POOL_SIZE

This parameter specifies the size of the Java pool, from which the Java memory manager allocates most Java state during runtime execution. This memory includes the shared in-memory representation of Java method and class definitions, as well as the Java objects that are migrated to the Java session space at end-of-call.

5. Conclusion

In this paper we have explained the memory architecture of one of the leading DBMS (i.e. Oracle 10g). It describes the instance configuration of oracle database server. It explains the detailed view of various parts of SGA (System Global Area) and the parameters used to fix and manipulate the memory for the various parts of SGA.

References

[1] Lightstone, S. *et al.*, "Toward Autonomic Computing with DB2 Universal Database", *SIGMOD Record*, Vol. 31, No.3, September 2002.

[2] Xu, X., Martin, P. and Powley, W., "Configuring Buffer Pools in DB2 UDB", IBM Canada Ltd., the National

Science and Engineering Research Council (NSERC) and Communication and Information Technology Ontario (CITO), 2002.

[3] Chaudhuri, S. (ed). Special Issue on, "Self-tuning Databases and ApplicationTuning", IEEE Data Engineering, *Bulletin 22(2)*, June 1999.

[4] Bernstein, P. *et al*., "The Asilomar Report on Database Research", ACMSIGMOD Record 27(4), December 1998, pp. 74 - 80.

[5] Nguyen, H. C., Ockene, A., Revell, R., and Skwish, W. J., "The role of detailed simulation in capacity planning". IBM Syst.J. 19, 1 (1980), 81-101.

[6] Seaman, P. H., "Modeling considerations for predicting performance of CICS/VSsystems", IBM Syst. J. 19, 1 (1980), 68-80.

[7] Foster, D. V., Mcgehearty, P. F., Sauer, C. H., and Waggoner, C. N., "A languagefor analysis of queuing models", *Proceedings of the 5th Annual Pittsburgh Modeling and Simulation Conference* (Univ. of Pittsburgh, Pittsburgh, Pa., Apr. 24-26). 1974, pp. 381-386.

[8] Reiser, M., and Sauer, C. H., "Queuingnetwork models: Methods of solution andtheir program implementation", *CurrentTrends in Programming Methodology*. Vol.3, Software Modeling and Its Impact onPerformance, K. M. Chandy and R. T. Yeb, Eds. Prentice-Hall, Englewood Cliffs, N. J., 1978, pp. 115-167.

[9] Borovits, I., and Neumann, S., "Computer Systems Performance Evaluation", D.C. Heath and Co., Lexington, Mass., 1979.

[10] Enrique Vargas, "High Availability Fundamentals", Sun Blue Prints™ OnLine, November 2000, http://www.sun.com/blueprints

[11] Harry Singh, "Distributed Fault-Tolerant/High-Availability Systems", Trillium Digital Systems, a division of Intel Corporation, 12100 Wilshire Boulevard, Suite 1800 Los Angeles, CA90025-7118 U.S.A. Document Number 8761019.12.

[12] David McKinley, "High availability system. High availability system platforms", Dedicated Systems Magazine- 2000 Q4 (http://www.dedicated-systems.com)

[13] Sasidhar Pendyala, "Oracle's Technologies for High Availability", Oracle Software India Ltd., India Development Centre

[14] James Koopmann, "Database Performance and some Christmas Cheer",an article in the Database Journal,January 2, 2003.

[15] Frank Naudé, "Oracle Monitoring and Performance Tuning", http://www.orafaq.com/faqdbapf.htm

[16] Michael Marxmeier, "Database Performance Tuning", http://www.hpeloquence.com/support/misc/dbtuning.html

[17] Sharma H., Shastri A., Biswas R. **"**Architecture of Automated Database Tuning Using SGA Parameters", *Database Systems Journal*, Romania, 2012

[18] Sharma H., Shastri A., Biswas R. "A Framework for Automated Database Tuning Using Dynamic SGA Parameters and Basic Operating System Utilities", *Database Systems Journal*, Romania, 2013

[19] Mihyar Hesson, "Database performance Issues"

[20] PROGRESS SOFTWARE, Progress Software Professional Services, http://www.progress.com/za/services/index.ssp

[21] Ralph Kimball, http://www.informatik.uni-trier.de/~ley/db/indices/atree/k/Kimball:Ralph.html

[22] Information Builders, "OLAP–Online Analytical Processing–Tools",

http://www.informationbuilders.com/olaponline-analytical-processing-tools.html

Hitesh KUMAR SHARMA, The author is an Assistant Professor in University of Petroleum& Energy Studies, Dehradun. He has published 10+ research papers in National Journals and 20+ research papers in International Journal. Currently He is pursuing his Ph.D. in the area of database tuning**.**

Aditya SHASTRI, Ph.D. MIT, Published about 230+ research papers in international journals on Graph Theory with applications in Communication, Computer Graphics and Parallel Processing, Vice Chancellor, Director, Banasthali University, Banasthali, INDIA

Ranjit BISWAS, Head and Professor Jamia Hamdard (Hamdard University) Published about150+ research papers in International journals/bulletins.

Sanjeev KUMARSINGH: The author is an Associate Professor in Galgotias University, Noida. He has published 35+ research paper in International Journal and 15+ research papers in National Journals. He is Ph.D. in Mathematics.

Analysis on the Metrics used in Optimizing Electronic Business based on Learning Techniques

Irina-Steliana STAN, Ion-Sorin STROE
University of Economic Studies, Bucharest, Romania
irina.stan@outlook.com, soso@soso.ro

The present paper proposes a methodology of analyzing the metrics related to electronic business. The drafts of the optimizing models include KPIs that can highlight the business specific, if only they are integrated by using learning-based techniques. Having set the most important and high-impact elements of the business, the models should get in the end the link between them, by automating business flows. The human resource will be found in the situation of collaborating more and more with the optimizing models which will translate into high quality decisions followed by profitability increase.
Keywords: *electronic business, learning systems, KPIs, optimization, performance, models, decision.*

1 Introduction

IT has been a component with major implications in the structure of all businesses even a century ago. Those companies that have treated this factor with high importance, by investing strategically, had a significantly higher rate of profitability in the long term under all internal departments.

Predictions for 2013 [1] show that IT spending will increase by 5.7%, exceeding the nominal amount of 2.1 billion dollars, due to the evolution of cloud technology, Big Data - from search to discovery and prediction (a classification, analysis and prediction of the data), specific platforms and social systems.

Having a developed market in financial terms, it is imperative that companies seek IT technologies in order to lead to the fulfillment of a plurality of objectives such as:

- Early detection and removal of socio-economic crisis, of system issues, of processes systematically affecting the company, penetrating to the highest level;
- Design an efficient and sustainable system, having a larger yield;
- Remove harmful processes and system gaps;
- Adapt business to changes independent from socio-economic environment.

In other words, in terms of investment in IT innovation, we distinguish three types of business:

- *Classical*. This category includes those businesses for which information technologies don't have a major influence, which gives the company a static kind by taking fundamental decisions in a slow pace;
- *Hybrid*. This includes all those companies that have discovered the advantages of using IT side, which try to automate and optimize by integrating informatics solutions. Their objective is that the technology should have a growing influence on decision-making systems;
- *Online*. Within these businesses, the main communications channel is online, integrating all technical innovations in the field.

Analyzing the evolution of the number of users of Internet services, there is a spectacular increase of 600% compared with 2000, meaning that in a population of about 7 billion, 2.5 billion frequently

accessed the Internet, having the penetration rate of 35% [2].
The statistics presented above reveal a new direction in terms of channel of communication between business and customers, so that companies that are open to new technologies should base their foundation in online environment.
It is worth mentioning that the online environment is a relatively new branch of information technology, the state of knowledge in an interdisciplinary approach, is at an average level, which is why research is needed in this area so as to develop integrated models to substantiate it.
Electronic business affairs do not eliminate the essential elements of classical businesses, but translate them so that models previously used in economics, finance, marketing and management, remain valid and used in the business in one way or another, the focus being on the information technology-based model.
Orientation on this topic leads to an increase in the average yield on long term, automatically on profitability too, the best examples coming from the analysis among the most profitable companies.
In the Top 100 Most Profitable Companies in 2012, the majority are those which adopted technologies or which exclusively perform activities in online environment, such as Apple (No. 3), Microsoft (No. 4), Intel (No. 12), Google (No. 18), Oracle (No. 23), Apache (No. 39) [3].
"One of the most important factors in the company is the technology" [4], according to a study made by IBM from over 1,700 company board of directors of the top companies, as presented in **Fig. 1**.

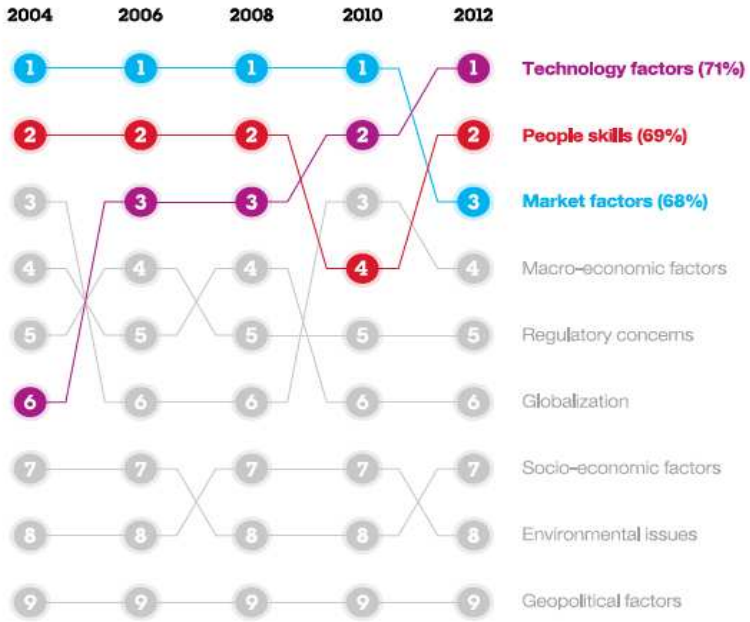


Fig. 1. "Technology pushes to the top of all the external forces that could impact their organization over the next 3 to 5 years, CEOs now see technology change as most critical." (Source: [4])

Electronic business can be divided into several categories, depending on the actors involved, on the technologies used, services offered, etc. But these topics will be discussed later, as the paper is focusing on the general area of this type of business.

Once specific technologies are integrated in the electronic business, their advantages should be used in a more efficient manner.
An important aspect is the one regarding business automation, in order to eliminate possible errors in a very short time or even predict them and get to take high-level decisions in the idea of building a decision

plan and a long term strategy that will ensure consistent services by positioning prior over competition. All must be able to perform in real time, to be flexible, intuitive and reversible, fault tolerant.

Starting point to build such a project is the data, whether these initially do not exist or are in low numbers.

One way to implement the above specifications can be by engaging a learning based system.

The first step is to train the learning system with a set of initial information in order to give a starting point, while, in parallel, the data providing process continues.

"Jerusalem Declaration" [5] shows the importance of current society in collecting data: "We are entering the era of a high rate of production of information of physical, biological, environmental, social and economic systems. The recording, accessing, data mining and dissemination of this information affect in a crucial way the progress of knowledge of mankind in the next years. Scientists should design, explore and validate protocols for the access and use of this information able to maximize the access and freedom of research and meanwhile protect and respect the private nature of part of it."

If in the past the collection of information supposed a major contribution of human capital by using expensive technology resources, now it can be obtained in a relatively short time, at a cost that would justify the investment.

Currently, obtaining important information that has major contributions can be made in real time due to interaction between several systems, by using their capacity of analysis and processing. As an example, more and more predictive models are used [6] with regards to economic future, election results and other socio-economic developments [7]. Another example is the "Google Dengue Trends" which uses information received from users based on their search on the search engine in order to determine the risk of disease by region due to mosquitos' stings [8].

Considering that the system is intended to be as general as possible, the need to cover vast areas of information makes the collected data diverse and heterogeneous. This leads to the need to build a set of minimum requirements in terms of data analysis and efficiency in processing because we aim to have a system which will obtain realistic results.

In the era of "Big Data" [9] there is no longer a question of how to obtain data but its analysis and processing. The International Data Corporation (IDC) shows that the data provided by users are exponentially growing, current information exceeding the storage capacity of 1.8 billion gigabytes, with a growth rate of 9 in less than five years [9].

The learning system must therefore include complex processing algorithm, which:

- Finds relevant information from many data sources;
- Makes a logical integration of information from different sources;
- Removes useless information;
- Focuses on continuous data homogenization, even on predefined steps;
- Assigns data and results to individual, giving a real meaning depending on the context;
- Provides an interface easily to integrate into other services which addresses to a wide variety of users so that their knowledge level does not represent an impediment in using the system.

The data collected at the moment can be classified according to many criteria but a special category is the human factor in terms of the contribution of this component in decision making and thus in making unwanted events.

The financial crisis of the past years was due to result of collective human behavior in the social environment on a large scale, which is why understanding individual typology based on historical information can facilitate similar items expectation of future crises disasters or other events.

Current technology provides an understanding of the human factor-based systems with very high accuracy, so as to design a complete system that would suggest strategies for future actions.
Stored data collection and analysis should:

- Determine pattern of collective actions in the social environment relying on a set of highly heterogeneous data;
- Define models of human behavior;
- Share responsibility for decisions, depending on cases;
- Determine initial points that favored specific actions;
- Show relationships between different models and other subjective but important factors such as human intuition, simulations of specific cases or test cases [10].

A very important thing to note is that we can automate many of the business processes using learning techniques on data but cannot remove the human factor, as it has the ability to determine alarm signals that the system might not be able to detect, which is why the objective is not to eliminate human intake but total optimization of the tasks that can be tech, leading to lower costs through dynamic adaptation to different situations and by replacing certain responsibilities that would be done on a longer period of time by individuals.

2. Overview on performance indicators

Term optimization involves the adoption of a mechanism that can be applied to electronic business and contains various internal processes that support adaptability to changes in a competitive context, fundamental decision-making dynamics, so, increasing profitability.
The main element of the optimization process must have as its starting point the possibility of measurement / analysis of all business flows.
The heterogeneity of different types of electronic business requires offering for adopting a generalized optimization solution which will homogenize the various input and output in order to get an easy to understand structure and adaptable to change, similar to the approach of Model Driven Architecture (MDA) [11]. According to this view, the end result is to get the definition, analysis, generalization and use of assessment tools sets for various categories of activities in the company.
MDA proposes an incremental framework that includes different theories such as:

- *The Goal - Question - Metric* (GQM) is a concept that assumes that measuring various indicators can be optimal only after the list of objectives is completed. Also, according to this principle, all information related to the objectives should be analyzed and used for recording the progress;
- *Balanced Scorecard* (BSC) is a strategic planning and management system used in various types of business that provides uniform internal activities to the vision and strategy, improving various impact factors;
- *Business Process Model and Notation* (BPMN) is a graphical representation of various business processes;
- *Semantics of Business Vocabulary and Business Rules* (SBVR) is a standard that defines the vocabulary and rules related to business terms, rules and procedures;
- Other theories.

Whatever core framework was used, proposed or already existing, the primary objective is not only to remain at the theoretical level but to materialize the optimization models by applying them on as many as possible electronic businesses.
The first step is to identify the key elements of the firm which suffer over time minimal or major changes. Thus, we use a system of metrics and indicators that will summarize the facts.

By definition, a metric can identify the deviation or potential deviation from achieving a goal set.

The main categories of indicators will be:

- *Key performance indicators (KPIs).* These will measure the changes in terms of returns on different time intervals. Performance indicators will express the quality of decisions implemented. Having the evolutionary history of the entity and of the performance indicators, they will identify any deviation that was not performance. Means of identification, correlation and automation of these indicators will be presented later on in this article;
- *Key Risk Indicators (KRIs).* Are those statistical indicators that can synthesize the positioning of the company in terms of risk, which are dynamically revised and aim to provide consistent signals on changes with high influence [12]. This type of indicators are the safety features to prevent incorrect application of decision strategies. Risk indicators will be able to determine the times when certain items will change and lead or can lead to behavior that could divert an aggravating behavior and could act in unfavorable directions.

Proposed sequence for defining the optimization platform in terms of metrics.

Step 1. Will review all elements defining the flow of business and will create a weighted matrix to express their importance. Also in this step will classify the main categories that can influence company's performance.

Step 2. Based on previously defined categories and on decisional factors, will express a number of key indicators for each stream of business.

Step 3. Each indicator will have detailed important information needed in the framework:

- The need for adoption, as in **Fig. 2.**;
- Whether or not an indicator has high impact, it will be classified in risk category indicators;
- The influence level between indicators;
- Reporting interval - hourly, daily, weekly, monthly, yearly, etc.;
- The period of storage and / or use of historical data;
- Range of monitoring the evolution of the index.

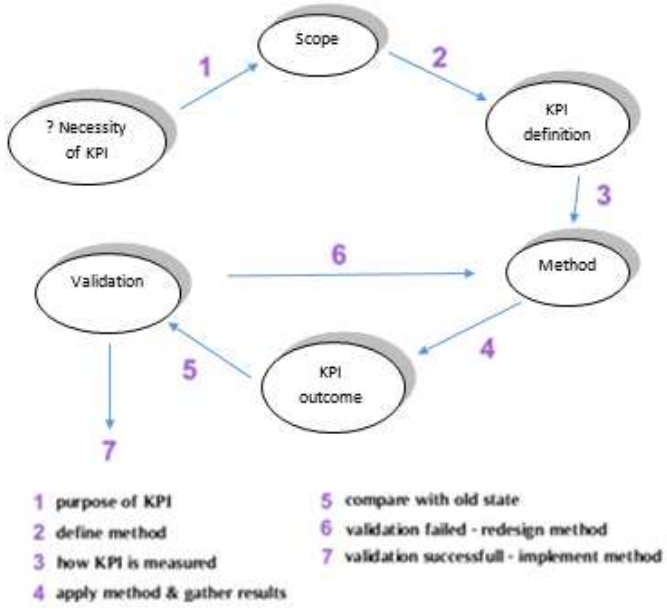


Fig. 2. Indicator analysis

Depending on the specific of category to be optimized, we will define optimization models that include previously defined indicators.

Fig. 3. illustrates the operation and application of two indicators.

Monitoring the risk state. Note that these performance indicators are at the same time risk indicators, which is why there are two levels - $MR(i_1)$ and $MR(i_2)$ - representing the values for indicators for which the risk status is triggered (Maximum Risk value). At the onset of risk status we will apply certain models of prevention and depending on how it will impact, will revert to the previous state. For example, if the indicator i_2 at time t_1 - $V(t_1, i_2)$ - triggers risk status, the core framework will manage time by an

eventual recovery of i_2 to the value recorded at time t_0.
Also, the complexity of the optimization algorithm will be determined in proportion with the influence of other risk indicators in creating the state.

- For i_1 and i_2 indicators, reporting intervals are different, weekly and monthly respectively, possibly due to their importance;
- It is worth mentioning that we are interested if the state of risk triggers only when the indicator is reporting. We can opt for a shorter reporting period but should take into consideration the nature of the indicator because changing the timeframe and fitting into alert level may be short-lived. Consequences can lead to misinterpretation of the indicator and beyond even taking wrong decisions. An example is highlighted in the chart below: the value V (t_3, i_2) is not included in the risk class, even if the index exceeds the maximum allowed by the next reporting period t_4.

Fig. 3. KPI evolution & evaluation

As a general conclusion on performance indicators, in order for them to bring a significant contribution to the optimization process they should be included in metrics whose utility will increase the yield rate and automatically improve profitability.

3. Optimization models for specific cases

The complexity of electronic business is similar to classical business, involving various fields, which combined create a flow system often difficult to track, control and predict. So, as part of the business, are elements related to the financial, economic, human, marketing, politics, etc.

The objective of this article is to take into account the business specific elements that develop their activity wholly or partly in the online environment, the reference categories being classified as follows:

- Software;
- Hardware;
- Human;
- Finance;
- Miscellaneous.

Optimization models to be proposed will be composed of metrics and indicators designed to partially or completely automate decision making system, having a dynamic opening, meaning fault-tolerant and adaptable. The goal is not to eliminate

the human factor but to provide assisted decision opportunities so that they act in time and understand the impact that decision might have.
Interaction between technology and the human factor will be an environment based on continuous improvement to ensure success. Only this approach will ensure efficient and early limiting of the human error by the monitoring performed on the computer system. Also, the individual will notice and correct possible deviations of the system before it produces results considered optimal while they are not.
In the following pages we propose models for the above mentioned categories.

3.1 Hardware

From the hardware point of view we can distinguish two subcategories, namely infrastructure, including communications channels, and the second one, of processing and storage of the resources.
As primary infrastructure can change based on the decisions taken by some important actors (large companies, governments, etc. - for example, all cars produced in the European Union have the obligation to incorporate the technology Anti-lock Braking System (ABS) [13]). Average users have low influence, which is why we will not discuss optimization techniques in this regard.
In terms of processing and storage of the resources, there is an adaptability of the market solutions according to the specific of the company. It assumes the use of Cloud Computing paradigm which allows scalability on infrastructure in a very short time (minutes). Increased storage and processing capacity correlated with lower price per unit allowed providing these general solutions but also personalized for each individual case [14], giving up the traditional solutions such as Virtual Private Server (VPS), Dedicated Servers, etc. [15].
Optimization will cover how hardware-solutions will be selected to meet the needs of online business platform and get a larger yield.
In terms of optimization, performance is perceived by users as high when the level of response from the platform is bigger, the maximum load being the instant one, in milliseconds. It is true that maintaining the performance level - level of satisfaction – involves other factors, which we will specify in other optimization categories, the main being software.
Website performance is a key factor in achieving user satisfaction and automatically converts them into active customers. At high level, Amazon shows [16] that a simple increase in the response time of just 100 milliseconds on the web platform results in decreased sales by 1%.
Hardware optimization goal is to obtain a model that:

- Minimizes the cost of hardware resources, by holding a level of satisfaction as high as possible;
- Justify the performance - response time per customer profitability.

Finally we get a relationship that will help us estimate the response time depending on profitability at a certain moment, taking into account historical data regarding various useful indicators such as:

Indicator	Description
ProcessingUnits	number of processing units used in Cloud for day X, month Y, year Z
StorageUnits	number of storage units used in Cloud for day X, month Y, year Z
TrafficUnits	number of traffic units used in Cloud for day X, month Y, year Z
VisitorNumber	number of platform visitors
CR	conversion rate = number of platform visitors who became clients/ total number of visitors
ClientIncome	income for each client
ResponseTime	response time (seconds) for each loaded page

Fig. 4. Miscellaneous hardware performance indicators

The model will serve to human part from the decision-making process in order to find answers based on existing information on questions such as "*What is the primary objective in terms of offered performance?*", "*What is the maximum cost that the company can support on the hardware infrastructure, when performances fall?*", "*What is the effect on limiting resources, meaning performance degradation, on the cost and the profitability per user?*".

The first step is to determine the causal relationship between *response time* (in seconds), meaning the *platform performance* and *profit per customer*.

Response time is based on indicators related to the number of sessions and requests in a given day, which involves a number of processing units, storage, traffic, etc.

Historical data modeling requires the use of an econometric model, the pattern resulting from the analysis of the number of requests for a certain period:

- *Requests per Day*: demand is not constant and is due to sociological factors. You can create oscillating trends driven by human factor availability to access platform, restrictions being imposed by the employer;
- *Requests per month*: in this case you can see a variable set of data, its nature being given by specific periods of each month, such as holidays, national events, etc.;
- *Requests per Year*: their number can vary due to political and economic actions, global or local, as a result of business performance.

Based on this analysis we conclude that the specific model is nonlinear. It requires using a sinusoidal model, starting from the general form:

$$Y_i = C + \alpha \sin(\omega T_i + \phi) + E_i$$

where:
C - is the constant defining the mean level;
α - is the wave amplitude for the model;
ω - is the frequency;
T_i- is time variable;
φ - is the phase;
E_i – is the error sequence in the sequence Y_i by approximating the model.

Translating into an econometric model for use, in this case, of historical data set, it will result in the following formula:

$$P = \sum_{i=m}^{n} (a_i * sin(b_i * RT + c_i) + E_i)$$

where:
P - is the estimation of customer profit, correlated with response time (RT);
m – is the starting year for the dataset involved in the model;
n – is the end year for the data set involved in the model;
RT – is the response time estimation, based on profit per customer (P);
a_i, b_i, c_i - model parameters used as training set;
E_i - is the error of approximation in the model.

Discovering the correlation between response time and profitability, human factor will have decision options in a range of project expenditures by adjusting the hardware infrastructure, namely the maximum number of processing units, storage and bandwidth used for a certain period.
Model results can be interpreted as a solution for a learning method. Basically, having the information available, the data is trained together with the links between them from time to time, the system being able to learn from previous periods and human decisions being taken so that ultimately the financial return is the maximum.

3.2 Software
The main element of this category is the online platform, the website, as an intermediary between business and customer flows.
Thus, performance indicators will be related to how online platform works and to most of the features coming from users. Indicators related to how to make and implement the site to make it functional, such as optimization by improving code written in a particular programming language or the use of specific utility programs, are connected to the infrastructure and can be included in the model from 3.1. We also exclude those indicators that can be included in the aforementioned category, as the demographic ones.
It may be created a variety of metrics [17] that can be used in an optimization model, for example:

- *Return of Investment* (ROI), conversions, subscribers;
- *Search Engine Optimization* (SEO): number of backlinks, quality of backlinks, pages, indexes, different ranks;
- *Usability*: time on site, bounce rate, links clicked, eyetracking, page visit, returning visitors;
- *Society*: on Twitter messages (tweets), Facebook likes.

In **Fig. 5.** we present three types of indicators that can justify the profitability per user in terms of software, the latter having a linear response correlated with the level of satisfaction. It is worth mentioning that a direct correlation between a variety of indicators and profitability can be found.

Indicator	Description
FunctionalityAccess	number of users that accessed the functionality in day X, month Y, year Z
FunctionalityIncome	number of users that become clients to that functionality in day X, month Y, year Z
UsedTechnology	a certain technology used by the visitor

Fig. 5. Miscellaneous software performance indicators

Choosing a linear model is motivated by the fact that the use of a given functionality is given by the customer satisfaction level for that utility.
The optimization model should assist the human in decision making process by answering questions such as *"What is the correlation between a specific user functionality and profit per user?", "How to reduce the cost of certain functionality by maintaining optimal levels of profitability?" "What consequences has removing a functionality within the financial structure?"*
The econometric linear model to correlate the functionality within the business and its profits will have the following form:

$$P = \sum_{i=m}^{n} (a_i * F + b_i + E_i)$$

where:

m – is the starting year for the dataset involved in the model;
n – is the end year for the data set involved in the model;
P - is the estimation of customer profit, correlated with accessing functionality (F);
F – is the estimation of accessing functionality, based on profit per customer (P);
a_i, b_i - model parameters used as training set;
E_i - is the error of approximation in the model.
Obtaining the correlation between functionality and profitability, human factor will get decision options in a range of project expenditures, by allocating funds for developing or disposing the service provided. The model determines the level of satisfaction of users.

4. Conclusions

In terms of optimization, it can extend across multiple categories and subcategories, including those ones already presented, financial, human, etc. Providing to the learning system a larger set of training data will give it the ability to almost completely automate decision-making process on matters of business. The quality of results on the learning process given from training the information will also be high as it is evidence-based and comprehensive.
Developing models presented in this article in future studies may lead to a refinement and determination of the specific of users, so that the optimization level will exceed expectations and investments will be made on items which confirmed the degree of profitability.

Acknowledgements

This work was co-financed by the European Social Fund through Sectoral Operational Programme Human Resources Development 2007-2013, project number POSDRU/159/1.5/S/134197 „Performance and excellence in doctoral and postdoctoral research in economic science from Romania”.

References

[1] Frank Gens (2013), *Competing on the 3rd Platform Opportunities at the Intersection of Mobile, Cloud, Social, and Big Data*, on-line: http://www.idc.com/research/Predictions13/downloadable/238044.pdf
[2] World Internet Users Statistics Usage and World PopulationStats (2013) , *Internet World Stats - Usage and Population Statistics.*, on-line: http://www.internetworldstats.com/stats.htm
[3] Fortune Magazine (2012), *Top companies: Most profitable*, on-line: http://money.cnn.com/magazines/fortune/fortune500/2012/performers/companies/profits/
[4] IBM Global Chief Executive Officer Study (2012), *Leading Through Connections*, on-line: http://public.dhe.ibm.com/common/ssi/ecm/en/gbe03485usen/GBE03485USEN.PDF
[5] Kahng Byungnam (2009), *The Jerusalem Declaration - Declaration on Data Access, Use and Dissemination for Scientific Research*, on-line: http://portale.unipa.it/ocs/home/jerusalemSubscription.html
[6] Sunstein Cass R. (2006), *Infotopia: how many minds produce knowledge*
[7] Abramowicz Michael (2009), *Predictocracy: market mechanisms for public and private decision making* ;
[8] GOOGLE (2013), *Google Dengue Trends - Dengue trends around the world.*, on-line: http://www.google.org/denguetrends/
[9] John Gantz, David Reinsel (2011), *IDC - Extracting Value from Chaos*, on-line: http://www.emc.com/collateral/analyst-reports/idc-extracting-value-from-chaos-ar.pdf

[10] D. Helbing, S. Balietti (2011), *From social simulation to integrative system design*, The European Physical Journal Special Topics, Volume 195, Issue 1, 69-100
[11] Gawel Bartlomiej, Skalna Iwona (2012), *Model Driven Architecture and classification of business rules modelling languages*, 2012 FEDERATED CONFERENCE ON COMPUTER SCIENCE AND INFORMATION SYSTEMS (FEDCSIS), 949-952
[12] Coleman (2009), Les. *Risk strategies*
[13] Europa (2007), *EC type-approval system for motor vehicles*, on-line: http://europa.eu/legislation_summaries/internal_market/single_market_for_goods/motor_vehicles/motor_vehicles_technical_harmonisation/n26100_en.htm
[14] Arutyunov V.V. (2010), *Cloud computing: Its history of development, modern state, and future considerations.*, Scientific and Technical Information Processing 39, no. 3, 173-178
[15] STROE SORIN (2011), *MySQL databases as part of the Online Business, using a platform based on Linux*, Database Systems Journal, vol. II, no. 3, 3-13
[16] Michele Mazzucco (2010), *Towards autonomic service provisioning systems*, IEEE International Symposium on Cluster Computing and the Grid, 273-282
[17] Sean Power (2010), *Metrics 101: What to Measure on Your Website*, Velocity Web Performance and Operations Conference

Irina-Steliana STAN graduated the faculty and master organized by the Faculty of Cybernetics, Statistics and Economic Informatics of the Academy of Economic Studies, with graduation and dissertation topic-themed works in database and management fields. She is currently a PhD candidate in the same area, Economic Informatics. Nowadays the interest domains are related to decision processes, learning systems and improvement on economical business.

Ion-Sorin STROE graduated the faculty and master organized by the Faculty of Cybernetics, Statistics and Economic Informatics of the Academy of Economic Studies, with graduation and dissertation topic-themed works in the electronic business field. He is currently a PhD in the same area, Economic Informatics. From the environmental point of view he presents Word Wide Web interest in new technologies and platforms, both conceptually, software and hardware too, developing over time many large and profitable web project.

Query Optimization Techniques in Microsoft SQL Server

Costel Gabriel CORLĂŢAN, Marius Mihai LAZĂR,
Valentina LUCA, Octavian Teodor PETRICICĂ
University of Economic Studies, Bucharest, Romania
gabi.corlatan@yahoo.com, lazar_mariusmihai@yahoo.com,
luca.valentina@ymail.com, octavian.petricica@ymail.com

Microsoft SQL Server is a relational database management system, having MS-SQL and Transact-SQL as primary structured programming languages. They rely on relational algebra which is mainly used for data insertion, modifying, deletion and retrieval, as well as for data access controlling. The problem with getting the expected results is handled by the management system which has the purpose of finding the best execution plan, this process being called optimization. The most frequently used queries are those of data retrieval through SELECT command. We have to take into consideration that not only the select queries need optimization, but also other objects, such as: index, view or statistics.
Keywords: *SQL Server, Query, Index, View, Statistics, Optimization.*

1 Introduction

We consider the following problems as being responsible for the low performance of a Microsoft SQL Server system. After optimizing the hardware, the operating system and then the SQL server settings, the main factors which affect the speed of execution are:

1. Missing indexes;
2. Inexact statistics;
3. Badly written queries;
4. Deadlocks;
5. T-SQL operations which do not rely on a single set of results (cursors);
6. Excessive fragmentation of indexes;
7. Frequent recompilation of queries.

These are only a few of the factors which can negatively influence the performance of a database. Further, we will discuss each of the above situations and give more details.

2. Missing indexes

This particular factor affects the most SQL Server's performance. When missing indexing of a table, the system has to go step by step through the entire table in order to find the searched value. This leads to overloading RAM memory and CPU, thus considerably increasing the time execution of a query. More than that, deadlocks can be created when for example, session number 1 is running, and session number 2 queries the same table as the first session.

Let's consider a table with 10 000 lines and 4 columns, among which a column named ID is automatically incremented one by one.

Table 1.1. Running a simple query to retrieve a row in a table

With clustered index (execution time / query plan)			**Without clustered index (execution time / query plan)**		
Time Statistics			Time Statistics		
Client processing time	5	→ 5.0000	Client processing time	5	→ 5.0000
Total execution time	327	→ 327.0000	Total execution time	327	→ 327.0000
Wait time on server replies	322	→ 322.0000	Wait time on server replies	322	→ 322.0000

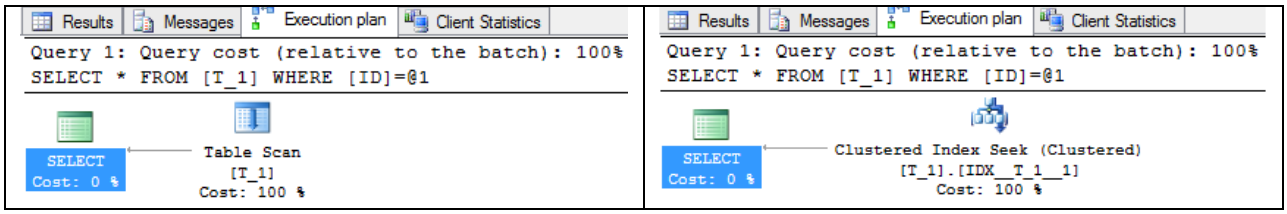
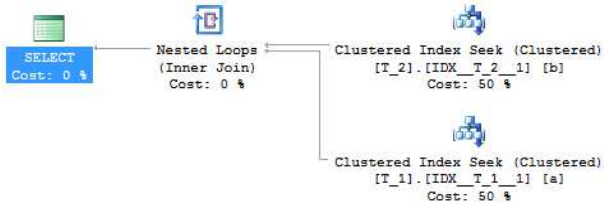
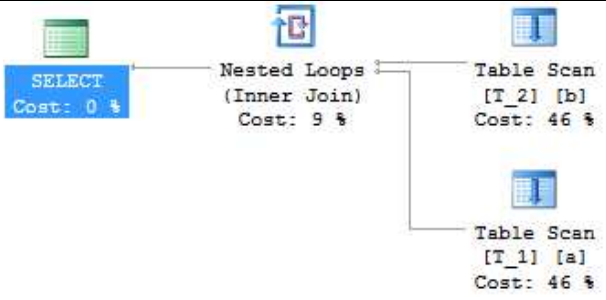


Table 1.2. Running a 2 table join query

With clustered index (execution time / query plan)	**Without clustered index (execution time / query plan)**
Time Statistics Client processing time 6 → 6.0000 Total execution time 16 → 16.0000 Wait time on server replies 10 → 10.0000	Time Statistics Client processing time 6 → 6.0000 Total execution time 16 → 16.0000 Wait time on server replies 10 → 10.0000
	

Tabel 1.3. Running a junction between two tables

Query (Q1)	**Query (Q2)**
`select*fromT_1whereID= 50000`	`select* fromT_1asa innerjoinT_2asb ona.ID=b.ID wherea.ID= 50000`

In Table 1.1, the query is created using a single table, with and without a clustered index on the column specified in the WHERE clause (Q1). In the second table (Table 1.2), the query has two tables, a join on ID column of the two tables and a WHERE clause (Q2).

According to [1] and [3], SQL Server supports the following types of indexes:

- Clustered index;
- Nonclustered index;
- Unique index;
- Columnstore index;
- Index with included columns;
- Index on computed columns;
- Filtered index;
- Spatial index;
- XML index;
- Full-text index.

According to [2], the main index optimization methods are the following:

- It is recommended that created indexes to be used by the query optimizer. In general, grouped indexes are better used for interval selections and ordered queries. Grouped indexes are also more suitable for dense keys (more duplicated values). Because the lines are not physically sorted, queries which run using these values which are not unique, will find them with a minimum of I/O operations. Ungrouped indexes are more suitable for unique selections and for searching individual lines;
- It is recommended for ungrouped indexes to be created with as low density as possible. Selectivity of an index can be estimated using the selectivity formula: number of unique keys/ number of lines. Ungrouped indexes with selectivity less than 0, 1 are not efficient and the optimizer will

refuse to use it. Ungrouped indexes are best used when searching for a single line. Obviously, the duplicate keys force the system to use more resources to find one particular line;

- Apart from increasing the selectivity of indexes, you should order the key columns of an index with more columns, by selectivity: place the columns with higher selectivity first. As the system goes through the index tree to find a value for a given key, using the more selective key columns means that it will need less I/O operations to get to the leaves level of the index, which results in a much faster query;
- When an index is created, transactions and key operations in database are taken into consideration. Indexes are built so that the optimizer can use them for the most important transactions;
- It is recommended that we take into consideration at the time of index creation, that they have to serve the most often combining conditions. For example, if you often combine two tables after a set of columns (join), you can build an index that will accelerate the combination;
- Give up the indexes which are not used. If, following the analysis of the execution plans of queries which should use indexes we see they cannot actually be used, they should be deleted;
- It is recommended creating indexes on references to external keys. External keys require an index with unique key for the referred table, but we have no restrictions on the table that makes the reference. Creation of an index in the dependent table can accelerate checking the integrity of external keys which result from the modifications to the referred table and can improve the performance of combining the two tables;
- In order to deserve the rare queries and reports of users, we recommend creating temporary indexes. For example, a report which is ran only once a year or once a semester does not require a permanent index. Create the index right before running the reports and give it up afterwards, if that makes things happen faster than running the report without any indexes;
- For unblocking page for an index, a system procedure can be used: *sys.sp_indexoptions*. This forces the server to use blocking at line level and table level. As long as the line blockings do not turn too often into table blockings, this solution improves the performance in the case of multiple simultaneous users;
- Thanks to using multiple indexes on a single table by the optimizer, multiple indexes with a single key can lead to a better overall performance than an index with a compound key. That is because the optimizer can query the indexes separately and can combine them to return a set of results. This is more flexible than using an index with compound key because the index keys on a single column can be specified in any combination, which cannot be done in the case of compound keys. Columns which have compound keys have to be used in order, from left to right;
- We recommend using Index Tuning Wizard application, which will suggest the optimized indexes for your queries. This is a very complex tool that can scan tracking files collected by SQL Server Profiler in order to recommend the indexes that will improve the performance.

3. Inexact Statistics

According to [3], the SQL Server database management system relies mostly on cost based optimization, thus the exact statistics are very important for an efficient use of indexes. Without these, the system cannot estimate exactly the number of rows, affected by the query. The quantity of data

which will be extracted from one or more tables (in the case of join) is important when deciding the optimization method of the query execution. Query optimization is less efficient when date statistics are not correctly updated.
The SQL Server query optimizer is based on cost, meaning that it decides the best data access mechanism, by type of query, while applying a selectivity identification strategy. Each statistic has an index attached, but there can be manually created statistics, on columns that do not belong to any index. Using statistics, the optimizer can make pretty reasonable estimates regarding the needed time for the system to return a set of results.

Indexed column statistics
The utility of an index is entirely dependent on the indexed column statistics. Without any statistics, the SQL Server cost-based query optimizer cannot decide which is the most efficient way of using an index. In order to satisfy this requirement, it automatically creates statistics on a index key every time the index is created. The required mechanism of data extraction in order to keep the cost low can use changing data. For example, if a table has a single row that matches some value which is unique, then using a nonclustered index makes sense. But if data changes, when adding a big number of rows with the same column value (duplicates), using the same index does not make any sense.
According to [5], SQL Server utilizes an efficient algorithm to decide when to execute the system procedure that updates the statistics, based on factors such as number of updates and table size:
- When inserting a line into an empty table;
- When inserting more than 1000 lines in a table that already has 1000 rows.

Automatic update of statistics is recommended in the vast majority of cases, except for very large table, where statistics updates can lead to slowing down or blocking the system. This is an isolated case and the best decision must be taken regarding its update.
Statistics update is made using the system procedure *sys.sp_updatestats* on an indexed table or view.

Unindexed column statistics
Sometimes there is the possibility of executing a query on an unindexed column. Even in this situation the query optimizer can take the best decision if it knows the data distribution of those columns. As opposed to index statistics, SQL Server can create statistics regarding the unindexed columns. Information regarding data distribution or the probability of having some value in an unindexed column can help the optimizer establish an optimum strategy. SQL Server benefits of query optimization even when it cannot use an index to locate the values. This automatically creates statistics on the unindexed column when the information that the system has, helps creating the best plan, usually when the columns are used in a predicate(ex: WHERE).

Table 1.4. Query plan

```
Query 1: Query cost (relative to the batch): 100%
select a.ID, a.Col_1, a.Col_2, b.Col_3 from T_1 as a inner join T_2 as b on a.ID = b.ID where a.ID = 50000
```

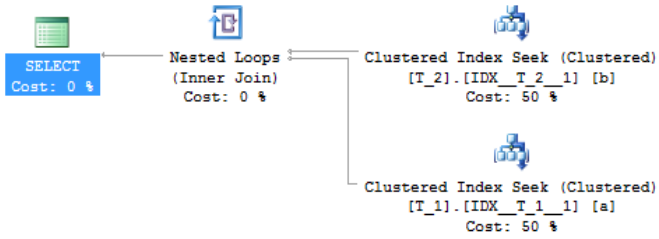


Table 1.5. Statistical data of the table "T_1"

```
selecta.ID,a.Col_1,a.Col_2,b.Col_3
       fromT_1asa
              innerjoinT_2asb
              ona.ID=b.ID
wherea.ID= 50000
```

4. Badly written queries

Index efficiency depends a lot on the way the queries are written. Taking a very large number of lines from a table can lead to inefficiency of the index. For improving performances, SQL queries must be written so that they use the existing indexes.

Table 1.6. T-SQL query to identify the names of the tables that contain at least one line using the system view "sys.partitions" (Q1)

```
select *
    from sys.tables as a with(nolock)
        inner join sys.schemas as b with(nolock)
        on a.schema_id = b.schema_id
        inner join sys.partitions as c with(nolock)
        on a.object_id = c.object_id
where a.type = 'U'
    and a.name like 'TI_MapCuvinte%'
    and b.name = 'dbo'
    and c.index_id < 2
    and c.rows > 0
```

Table 1.7. T-SQL query to identify the names of the tables that contain at least one line using the system view "sys.sysindexes" (Q2)

```
select *
    from sys.tables as a with(nolock)
        inner join sys.schemas as b with(nolock)
        on a.schema_id = b.schema_id
        inner join sys.sysindexes as c with(nolock)
        on a.object_id = c.id
where a.type = 'U'
    and a.name like 'TI_MapCuvinte%'
    and b.name = 'dbo'
    and c.indid < 2
    and c.rows > 0
```

Table 1.8. Query execution time using the system view "sys.partitions"

Time Statistics			
Client processing time	50	→	50.0000
Total execution time	4137	→	4137.0000
Wait time on server replies	4087	→	4087.0000

Table 1.9. Query execution time using the system view "sys.sysindexes"

Time Statistics			
Client processing time	46	→	46.0000
Total execution time	11702	→	11702.0000
Wait time on server replies	11656	→	11656.0000

As an example we select two T-SQL inquiries executed on system tables (views). Both return the names of the table which start with “TI Word Map” from the “dbo” layout and which contain at least one line. This method is more efficient than rolling a slider key on all tables from “sys.tables”, than rolling an inquiry “select count(*) from table name”, for each line with slider key, insertion of tables which contain at least one line in a temporary table, and then its inquiry. However this number can be determined by two methods: Table 1.6 and Table 1. 7. Although they may be identical the only difference between the two inquiries is that for determining the number of lines in Q1 table it extracts the number from sys.partitions and for Q2 from sys. sysindexes. As we may see Q1 is aproximatively three times quicker than Q2. In this case it is recommended using the system view sys.partitions rather than sys. sysindexes which according to Microsoft will be erased in future versions of SQL Server data bases.

Methods for optimizing SELECT option:

- Every time possible it is recommended to use as search columns in inquiries, the far left ones of the index. One index on col_1 and col_2 is of no help in an inquiry which filtrates results of col_2;
- It is recommended to build up WHERE terms which inquiry optimizer should recognize and use as searching tools;
- Don’t use DISTINCT or ORDER BY without any need. They may be used only to eliminate duplicate values or to

select a specific order in the result set. With the sole exception when the optimizer can find an index that might serve them, they can engage an intermediate working table, which can be expensive when talking about performance;

- Use UNION ALL instead of UNION when eliminating duplicates from a result set is not a priority. Because it eliminates the duplicates, UNION must sort or deal the result set before returning it;
- You may use SET LOCK_TIMEOUT when controlling the time limit a connection is waiting for a blocked resource. At the start of the session the automatic variable @@LOCK_TIMEOUT returns -1 which means that no value was selected to expire. You can select as a value for LOCK_TIMEOUT any positive number which establishes the number of milliseconds which an inquiry waits a blocked resource before to expire. In more difficult stages this is necessary to prevent apparent blocked applications;
- If an inquiry includes the IN predicate which contains a list of constant values (instead of a minor inquiry) order the values according to the release frequency in the exterior inquiry, more over when you know data tendencies. A common solution is alphabetical or numerical ordering of values but these may not be optimal. Because the predicate returns TRUE as soon as it reaches a resemblance for any of its values, moving on the first positions of the list the values which are released frequently should accelerate the inquiry, especially in the case when the column where the searching is done, is not indexed;
- It is recommended choosing algorithms despite of imbricated minor inquiries. A minor inquiry may need an imbricated inquiry that is a cycle in a cycle. In the case of imbricated error, the lines of the interior table are scanned for each line of the exterior table. This thing works very good for little tables and it was the only algorithm strategy used in SQL Server before 7.0 version, but as tables become bigger and bigger this solution becomes less and less efficient. It is much better to do the normal algorithms between tables and to let the optimizer select the best way to analyze them. Mostly the optimizer will try to transform the pointless minor inquiries in algorithms;
- When possible it is recommended to avoid CROSS JOINT type algorithms. With the exception of the case in which one cannot avoid the need for Cartesian product of two tables, it is used a more efficient method of algorithms to chain one table after the other. Returning an unwanted Cartesian product and then eliminating the duplicates generated by it using DISTINCT and GROUP BY is a problem which causes serious damage to the inquiry;
- You may use TOP(n) extension to restrict the number of lines returned by an inquiry. This thing is useful mainly when you may insert values using SELECT, because you may view only the values from the first part of the table;
- You may use OPTIONS clause of the SELECT instruction for influencing the inquiry optimizer with inquiry suggestions. As well you may select suggestions for tables and specific algorithms. As a rule the optimizer should optimize the inquiries, but there are cases in which the performing plan chosen is not the best. Using suggestions for inquiry, table or algorithm, you may entail to a certain type of algorithm, group or union to use a certain index, so on and so forth. These are called query hints; Here are some of these:
    - FAST number_of_lines – points out that the inquiry is optimized

to rapidly reclaim the first "number of lines" After the first "number of lines" are returned, the inquiry goes on and produces the complete result set;
  - FORCE ORDER – points out that the syntax order of the inquiry is enabled during the inquiry optimization;
  - MAXDOP processor_number – overwrites the maximum number of configurated parallelism using sp_configure;
  - OPTIMIZE FOR (@variable_name) – points out to the inquiry optimizer to use a specific value to a certain local variable when inquiry is compiled and optimized;
  - USE PLAN – impels the inquiry optimizer to use an existent inquiry plan. It can be used with: insert, update, merge and delete options.
- Beside query hints there are also table hints, which influence the inquiry optimizer in taking some decisions as: using a blocking method for a table, using a certain index, blocking lines, etc. Here are some types:
  - NOLOCK – READUNCOMITTED and NOLOCK hints are applied only when data is blocked. They obtain Sch-s (stability scheme) blocked when compelling and executing. This indicates no blocking and doesn't stop other transactions accession to data, including their modification;
  - INDEX – impels using an index;
  - READPAST – points out to the system not to read the lines which are blocked by other transactions;
  - ROWLOCK – a line is blocked;
  - TABLOCK – points out that the blocking is at a table level;
  - HOLDLOCK – it's the equivalent of the SERIALIZABLE isolation level;
  - TABLOCKX – points out an exclusive blocking of the table.
- Testing and comparing two inquiries to determine the most efficient way of accessing data, one must be sure that the mechanism of data placement in cache memory of the SQL Server doesn't impair test results. One way of doing this is by cycling the server between inquiries rolling. Another way is by using undocumented DBCC commands to clean important cache memories. DBCCFREEPROCACHE extricates the memory for the procedure. DBCC DROPCLEANBUFFERS cleans all cache memories.

5. Excessive blocking (deadlocks)

According to [2], SQL Server is a software compliant with "ACID" rules where "ACID" is an acronym for atomicity, consistency, isolation and durability. This assures that the variations made by the simultaneous transactions are worthy isolated one versus the other. Compliance with the rules by the transactions is mandatory for data safety.

- ***Atomicity*:** one transaction is atomic if compliant with the principle "all or nothing". When a transaction is successful all its variations become permanent, when a transaction fails, all the variations are canceled;
- ***Consistency*:** one transaction is atomic if compliant with the principle "all or nothing". When a transaction is successful all its variations become permanent, when a transaction fails, all the variations are canceled;
- ***Isolation*:** a transaction is isolated if it doesn't concern other transactions or it isn't concerned by other rival

transactions on same data (levels of isolation);
- ***Durability:*** a transaction is durable if it can be done or reversed despite of a system breakdown.

SQL Server Blocking

When a session is doing an inquiry, the system determines the resources of the data base and gives data base blocking for the respective session. The inquiry is blocked in case that a session got blocked. Despite all these, for offering isolation and rivalry SQL Server offers different levels of blocking as:

- *Row (RID***)** – this blocking is done on a single row in a table and is the smallest blocking level. For example when an inquiry changes a row in a table a RID blocking is made for that inquiry.
- *Key (KEY)* – this is a blocking system of a row in an index and is identified as a blocking key. For example for a table with a clustered index, data pages of the table and data pages of the clustered index are the same. Since both rows are the same for the clustered index table only a KEY type blocking is obtained on the clustered index row or the limited set of rows while accessing the rows in the table;
- *Page (PAG)* – PAG blocking system is kept only on a page of a table or index. When an inquiry requires more rows in a page, the consistency of all required rows can be kept whether by RID or KEY blocking on a row level or by PAG blocking on a page level;
- *Extent (EXT***)** – this type of blocking is made after using ALTER INDEX REBUILD command. This command is used at a table level and the table pages can be moved from one scale to another scale. All this time the integrity of the extent is protected by EXT blocking;
- *Heap or B-tree (HoBT***)** – a HEAP or B-TREE blocking is used when data is heaped on many file groups. Target object may be a table without clustered index or a B-TREE object. A setting in ALTER TABLE offers a certain level of control over the blocking mode (lock escalation). Because the parts are heaped in many file groups, each has to have its own data assignation definition. HoBT blocking type acts on a partition level, not on a table level;
- *Table (TAB)* – this blocking type is the highest blocking level. Books complete access to the table and its indexes;
- *File (FIL)*;
- *Application (APP)*;
- *MetaData (MDT)*;
- *Allocation Unit (AU)*;
- *Database (DB)* – it is a blocking system kept on a database level. When an application gets access to a data base, the blocking manager offers a blocking at a data base level worthy to a SPID (Speed Process ID).

Table 1.10. The session (event session) for a database jams

```
if exists(select top(1) 1 from sys.server_event_sessions where name='Session_GasesteBlocaje')
begin
    drop event session Session_GasesteBlocaje ON SERVER
end
go

declare @dbid int
set @dbid = (select db_id('bdsa_optimizare'))

if(@dbid is null)
begin
    raiserror('Eroare', 18, 1)
    return
end

declare @sql    nvarchar(max)
set @sql =
'
CREATE EVENT SESSION Session_GasesteBlocaje ON SERVER
ADD EVENT sqlserver.lock_acquired
    (action
        (sqlserver.sql_text, sqlserver.database_id, sqlserver.tsql_stack, sqlserver.plan_handle, sqlserver.session_id)
    where (database_id=' + cast(@dbid as nvarchar) + ' AND resource_0 != 0)
    ),
ADD EVENT sqlserver.lock_released
    (where (database_id=' + cast(@dbid as nvarchar) + ' AND resource_0 != 0))
ADD TARGET package0.pair_matching
    (set begin_event=''sqlserver.lock_acquired'',
            begin_matching_columns=''database_id, resource_0, resource_1, resource_2, transaction_id, mode'',
            end_event=''sqlserver.lock_released'',
            end_matching_columns=''database_id, resource_0, resource_1, resource_2, transaction_id, mode'',
            respond_to_memory_pressure=1)
WITH (max_dispatch_latency = 1 seconds)
'
    --print(@sql)
    exec(@sql)

    --start;
alter event session Session_GasesteBlocaje on server
state = START
```

Isolation levels of transactions

SQL Server accepts four types of isolation of a transaction. As I mentioned earlier the level of isolation of a transaction controls the way in which it affects and is affected by other transactions. In a level of isolation it is always a reverse relation between data consistency and users rivalry. Selecting a more restrictive level of isolation amplifies data consistency to the detriment of accessibility. Selecting a less restrictive level of isolation amplifies rivalry to the detriment of data consistency. It is important to balance these opposite interests, so the needs of the application to be assured.

- *READ UNCOMMITTED* - READ UNCOMMITTED parameter specification is essentially the same as using NOLOCK option in each table referenced by a transaction. This is the least restrictive of the four isolation levels in SQL Server. It allows "dirty reads" (reading not completed changes of other transactions) and readings that cannot be repeated (data which changes between readings during a transaction);
- *READ COMMITTED* - The default isolation level in SQL Server, so if you do not specify otherwise, you get the READ COMMITTED level. READ COMMITTED level avoids reading "dirty", enforcing shared locks on accessed data but allows modification of basic data during a transaction, thus the possibility of repeated readings and / or phantom data;
- *REPEATABLE READ* - The REPEATABLE READ level generates locks that prohibit other users to modify data accessed by a transaction, but does not prohibit the insert of new rows, which may result in phantom rows between readings during a transaction;
- *SERIALIZABLE* - The level SERIALIZABLE prevents "dirty reads" and phantom lines by

introducing key-range locks on data access. It is the most restrictive of the four isolation levels in SQL Server. This is equivalent to using the option HOLDLOCK on each table referenced by a transaction.

In order to enforce an isolation level for a transaction, the command SET TRANSACTION ISOLATION LEVEL must be used. The valid parameters for the isolation levels are READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ and SERIALIZABLE.

6. T -SQL operations that are not based on a single set of results (cursors)

Given that Transact -SQL is a set based language, we know that it operates on sets. Writing non-set based code can lead to excessive use of cursors and loops. To improve performance, it is recommended to use queries based on set, not the row by row approach, as the latter leads to hardware overloading, resulting in a much higher data access time. Excessive use of cursors increases the stress on SQL server and results in reduced system performance.

Classification cursors

Based on the level of isolation, according to [3] and [4] cursors may be:

- ***Read-only*** – the cursor cannot be updated;
- ***Optimistic***– updatable cursor and uses the optimistic concurrency model (does not lock rows of data);
- ***Scroll Locks*** – updatable cursor that has a locking system for any row to be updated.

Types of cursors

- ***Forward-only*** – the default cursor type. It returns the rows from the data set sequentially. No extra space needed in tempdb and changes on data are visible as soon as the cursor reaches it;
- ***Static*** – static cursors return a result set that can only be read, which is impervious to changes in data. They are sometimes referred to as insensitive and are the opposite of dynamic cursors;
- ***Dynamic*** – as with forward- only cursors, dynamic cursors type reflect changes on rows as they reach them. No extra space in tempdb is needed, and unlike forward-only cursors they are scrollable, which means they are not restricted to sequential access on rows. Sometimes these cursors are called sensitive;
- ***Keyset*** – keyset cursors return a fully scrollable result set, whose membership and order are fixed. As with static cursors, the unique-key values set of the cursor lines is copied to tempdb when opening the cursor. This is why the cursor membership is fixed.

Cost comparison

Read-Only cursors

The read-only model has the following advantages:

- The lowest level of locking: the read-only model introduces the lock with the least impact and database synchronization. Locking on the base row while the cursor loops the rows can be avoided by using the NO_LOCK command in the SELECT instruction, but the dirty reads must be taken into account;
- The highest level of concurrency: as supplementary locks are not held in the rows that form its base, the read-only cursor does not block other users from accessing the base table.

The main disadvantage of the read-only cursor refers to the lack of updates: contents of the base table cannot be modified by the cursor.

Optimistic cursors

The optimistic model has the following advantages:

- Low level of locking - Similar to the read-only model, the optimistic concurrency model does not have a specific type of lock. In order to further improve the concurrency, the NOLOCK locking instruction may also be used, as in the case of the read-only concurrency model. Using the cursor to update a row requires exclusive rights on that row;
- High concurrency - Since only a shared lock is used on the rows behind it, the cursor does not block other users from accessing the base table. Changing a baseline will block other users from accessing the row during the update.

Scroll Locks cursors

The scroll locks model has the following advantages:

- Concurrency control: By blocking the base row corresponding the last row of the cursor, it ensures that the base row cannot be changed by another user;
- For very large data sets, the use of asynchronous cursors is recommended. Returning a cursor allows further processing while the cursor is populated. Asynchronous cursors are set as follows:

```
EXECsp_configure'show
advanced options', 1;
GO
RECONFIGURE
GO
EXECsp_configure'cursor
threshold', 0;
GO
RECONFIGURE
GO
```

- When configuring read-only result sets, unidirectional, FAST_FORWARD cursor option is recommended instead of FORWARD_ONLY. FAST_FORWARD option creates a FORWARD_ONLY and a READ_ONLY cursor with several built in performance optimizations;
- It is recommended to avoid changing a large number of lines using a cursor cycle contained in a transaction because each line that it changes may remain locked until the end of the transaction, depending on the transaction isolation level.

7. Excessive index fragmentation

Normally, data is organized in an orderly way. However, in the case in which the pages contain fragmented data or contain a small amount of data due to frequent page division, the number of read operations needed to return the data will be higher than usual. Increasing the number of readings is due to fragmentation.

Fragmentation occurs when table data is changed .When executing instructions to insert or update data (INSERT or UPDATE statements), clustered indexes on tables and nonclustered indexes are affected. This issue can cause a tear in the index leaf page when an index change cannot be stored on the same page. A new leaf page will then be added as part of the original page in order to maintain the logical order of rows in the index key. Although the new leaf page maintains the logical order of rows of data from the original page, it usually will not be physically adjacent to the original page on disk. For example, suppose that an index has nine key values (or rows in the index) and average index rows size allows a maximum of four rows in a leaf page. 8KB leaf pages are connected to previous and next pages to maintain the logical order of the index In Figure 1.1, leaf pages layer 1 is shown. Since the key values of index leaf pages are always sorted, a new row of the index with a key value of 25 must occupy a place between the extended key values of 20 and 30. Since the leaf page containing that existing index is filled with four rows, the new index row will result in dividing the page.

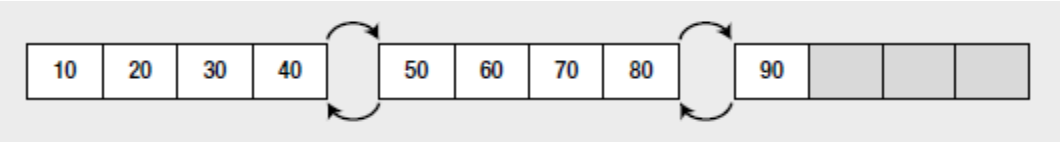


Fig 1.1. Layer leaf pages

A new page will be assigned to the index and a part of the first page will be moved on this new page so that the new index key can be expressed in the correct logical form. The links between indexed pages will also be updated as the pages are logically linked in index order. As it can be seen in Figure 1.1, a new page, even if linking it to the other pages id done in the correct logical order, physically the linking can sometimes be unordered.

Pages are grouped into larger units called extents, which may include eight pages. SQL Server uses unit as a physical unit of disk allocation. Ideally, the physical order of extensions contains leaf pages whose key must be the same as the logical order of the index. This reduces the number of required switches between extents when reading a series of rows in the index. However, crevice in the pages can disturb the pages within extensions physically and can also cause physical disorder even among extensions. For example, suppose the first two leaf pages of the index are as measure 1 and the third page is measure 2. If measure 2 contains unallocated space, then the new leaf page allocated to the index causes page division. The page will be measure 2, as shown in Figure 1.3.

With leaf pages distributed between two extents, one would ideally expect a read of a series of index rows with a maximum of one switch between the two extents. However, the disorganization of pages between extents can cause more than one switch during the read of a series of rows in the index. For example, to retrieve a number of rows in the index between 25 and 90, you will need three switches between the two extents, as follows:

- Firstly, a measure switch to retrieve the value of key 30 after the value of key 25;
- Secondly, a measure switch to retrieve the value of key 50 after the value of key 45;
- Thirdly, a measure switch to retrieve the value of key 90 after the value of key 80.

This type of fragmentation is called external fragmentation. Fragmentation can also occur in an index page. If an INSERT or UPDATE operation creates a page break, then free space will be left behind in the original leaf page. Free space can also be caused by a DELETE operation.

The net effect is the reduction of the number of rows included in one leaf page. For example, in Figure 1.3, the page split caused by the INSERT operation has created a gap in the first leaf page. This is known as internal fragmentation.

For a highly transactional database, it is preferable to deliberately leave space inside the leaf pages to be able to add new rows or change existing rows size without causing a rift in the page. In Figure 1.3 the free space of the first leaf page allows a key value in the index of 26. This can be added to the leaf page without causing a rift.

Heap pages can become fragmented in exactly the same way. Unfortunately, because of the storage mechanism and because every nonclustered index uses the physical location of the data to retrieve data from heap, defragmenting is quite difficult. The ALTER TABLE command using the REBUILD clause can be used to perform a reconstruction.

SQL Server 2012 exposes leaf pages and other data through a dynamic system view management system called sys.dm_db_index_physical_stats. It stores both index size and degree of fragmentation.

According to Microsoft, the commands ALTER INDEX REORGANIZE and ALTER INDEX REBUILD are used depending on the degree of fragmentation of the system view as follows:

Table 1.11. Degree of fragmentation of the system

Value of avg_fragmentation_in_percent column	**T-SQL code**
> 5% and <= 30%	ALTER INDEX REORGANIZE
> 30%	ALTER INDEX REBUILD

Rebuilding an index can be performed either online or offline, but reorganization can only be executed online.

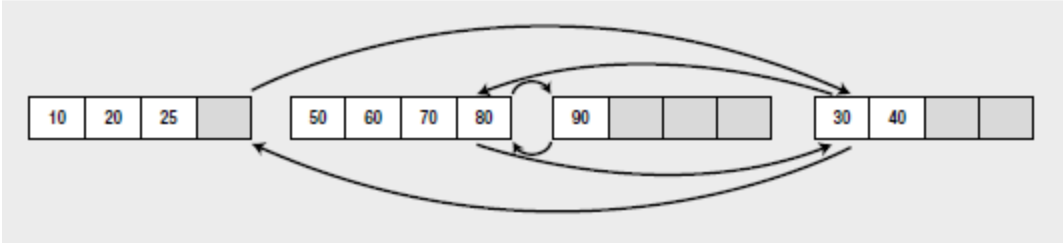


Fig 1.2. Outside leaf page order

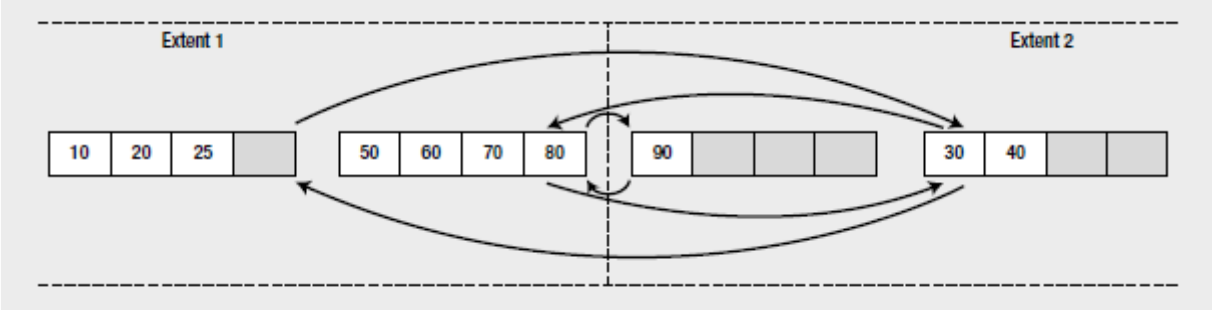


Fig 1.3. Outside leaf pages order distributed extensions

Note that this index fragmentation is different from disk fragmentation. Index fragmentation can't be determined by running the Disk Defragmenter tool, because the order pages in a SQL Server file is understood only by SQL Server, not by the operating system.

```
select b.name, a.name, a.type_desc, d.avg_fragmentation_in_percent
    from sys.indexes as a with(nolock)
        inner join sys.tables as b with(nolock)
        on a.object_id = b.object_id
        inner join sys.schemas as e with(nolock)
        on b.schema_id = e.schema_id
        left join sys.xml_indexes as c with(nolock)
        on a.object_id = c.object_id and a.index_id = c.index_id
        left join sys.spatial_indexes as f with(nolock)
        on a.object_id = f.object_id and a.index_id = f.index_id
        cross apply sys.dm_db_index_physical_stats(DB_ID(DB_NAME()), null, null, null, 'DETAILED') as
where a.object_id = d.object_id and a.index_id = d.index_id
    and e.name = 'dbo'
    and b.name != 'sysdiagrams'
    and d.avg_fragmentation_in_percent > 10
```

Fig 1.4. Determining the degree of fragmentation of the index tables in schema "dbo" with fragmentation greater than 10

	name	name	type_desc	avg_fragmentation_in_percent
1	T_4	IDX__T_4__1	CLUSTERED	50
2	T_5	IDX__T_5__1	CLUSTERED	66.6666666666667
3	T_6	IDX__T_6__1	CLUSTERED	58.3333333333333

Fig 1.5. The result of the query to determine the degree of fragmentation of indexes

8. Rebuilding Frequent Queries

The most common way to provide a reusable execution plan, independently of the variables used in a query, is to use a stored procedure or parameterized query. By creating a stored procedure to execute a set of SQL queries, the database system creates a parameterized execution plan independently of the parameters during execution. The execution plan generated will be reusable only if SQL Server does not have to recompile individual statements from stored procedure each time it is executed (e.g. sequences of dynamic SQL). Rebuilding frequent query execution causes time increases.

Optimizing stored procedures methods:

- Whenever possible it is recommended to use stored procedures instead of ad-hoc queries. In order to reuse the execution plan of an ad hoc SQL query you have to match exactly and must fully qualify each object meant. If in future use the query, everything is different: the parameters, name objects, key elements of SET, the plan will not be reused. A good solution that avoids the limitations of ad-hoc queries is to use the system stored procedure sys.sp_executesql. This is somewhere between rigid stored procedures and ad hoc Transact-SQL queries, allowing to run ad-hoc queries with replaced parameters. This facilitates reuse of ad-hoc execution plans without the need for precise consistency;
- For a small portion of a stored procedure the query plan must be rebuilt at every execution (e.g. due to data changes that doesn't make the optimal plan), but we do not want the overload associated with rebuilding plan for the entire procedure each time, that portion it should be moved in a stand-alone procedure. This allows reconstruction of its execution plan every time a run, but without affecting the procedure longer. If this is not possible, try using EXEC() to call the suspect code in the main procedure. Because this subroutine it is dynamically built we can generate a new execution plan at every execution, without affecting the whole stored procedure query plan;
- When possible, it is better to use output parameters of stored procedures instead of set results. If you need to return the

result of a calculation or to locate a single value in a table, it is preferable to return the output parameter of a stored procedure instead of a set result with a single line. Even if you return multiple columns, output parameters of stored procedures are more effective than complete set results;

- When you need to return a set of lines from a stored procedure to another, it is better to use output parameters of the cursor instead of set results. This technique is considerably more flexible and allows the second procedure to run more quickly since it doesn't work as set results. Then the caller can process the rows returned by the cursor as desired;
- It is recommended to minimize the number of network packets between the client and server. A very effective way to achieve this goal is to disable DONE_IN_PROC messages. You can disable it at the procedure level with the SET NOCOUNT command or at the server level with tracking indicator 3640. Doing so may lead to huge differences in performance, especially when relatively slow networks are used such as WAN networks. When you choose not to use the tracking indicator 3640, SET NOCOUNT ON should be used at the beginning of any stored procedures that you write;
- When adjusting query use PROCCACHE DBCC command to list information about cache memory reserved for the procedure. Also use the DBCC FREEPROCCACHE command to clear the memory cache so that multiple executions of a given procedure not alter test results. DBCC FLUSHPROCINDB used to force the creation of new execution plans for basic procedures.

Conclusions

Performance optimization is an ongoing process. This process requires continuous monitoring and improving database performance. The purpose of this paper is to provide a list of SQL scenarios to serve as a quick and easy reference guide during the development phase and maintenance of the database. Transact-SQL language provides a wide range of techniques for updating query. On the top of the list are found correct database design, adding indexes and query interrogations. Performance optimization is a complex subject, which certainly could fill many books. The secret to success is knowing the instruments mentioned above in relation to the available space, knowing how the server works and the own the required skills to solve problems using this knowledge.

References

[1] Adam Jorgensen, Jorge Segarra, Patrick Leblanc, Jose Chinchilla, Aaron Nelson, *Microsoft SQL Server 2012 Bible*, Ed. Johs Wiley & Sons, Inc., 2012, Indianapolis, Indiana - USA, ISBN: 978-1-118-10687-7

[2] Ken Henderson, *Transact-SQL (*Titlul original: *The Guru's Guide to Transact-SQL)*, Ed. Teora, București, România, 2002, ISBN: 973-20-0612-9

[3] Grant Fritchey, *SQL Server 2012 Query Performance Tuning*, Ed. Apress, USA, 2012, ISBN: 978-1-4302-4203-1

[4] Leonard Lobel, Andrew Brust, *Programming Microsoft SQL Server 2012*, Ed. Microsoft Press, 2012, ISBN-13: 978-0735658226

[5] Jason Strate, TedKrueger, *Expert Performance Indexing for SQL Server 2012*, Ed. Apress, USA, 2012, ISBN: 978-1-4302-3741-9

Costel Gabriel CORLĂŢAN studies at Academy of Economic Studies, Database for Business Support master program. Main technologies that he is working with are: MS SQL Server, .NET Framework, ASP. NET, C#.

Marius Mihai LAZĂR: Database for Business Support masterand and programmer in the area of database developing, tuning and optimization, mainly Microsoft technologies like SQL Server and Visual Studio. Also he develop web, desktop and mobile applications. In 2012 he published and documented an application about “Elementary cellular automaton”.

Valentina LUCA graduated from the Faculty of Cybernetics, Statistics and Economic Informatics of the Bucharest Universisty of Economic Studies in 2012. Currently she is a masterand enrolled in Databases for Business Support program. Her fields of interest: databases, data warehouses, business intelligence.

Octavian Teodor PETRICICĂ: graduated from the Faculty of Mathematics and Informatics of the Hyperion University of Bucharest in 2012. He is currently a masterand of Database for Business Support program. His scientific fields of interest include: Databases, Web development and Application programming interfaces. At present he is a Software Developer in the department of IT Applications at Cargus International SA.

The Transition from RDBMS to NoSQL. A Comparative Analysis of Three Popular Non-Relational Solutions: Cassandra, MongoDB and Couchbase

Cristina BĂZĂR, Cosmin Sebastian IOSIF
University of Economic Studies, Bucharest, Romania
cristina.bazara@gmail.com, iosifsebastian@yahoo.com

NoSQL databases were built in the need to deal with the increasing amount of complex data (Big Data), required in real-time web applications, and are mostly addressing some of these points: the focus on availability over consistency, horizontally scalable, distributed architecture, and open-source. The purpose of this paper is to present the reasons for a transition from RDBMS to NoSQL databases, to describe the main characteristics of non-relational databases and to compare and analyze three popular NoSQL solutions – Cassandra, MongoDB and CouchBase, outlining the results obtained during performance comparison tests. Each solution is optimized for different workloads and different use cases. Therefore, each has its own strong points and weaknesses.
Keywords: *NoSQL, Relational vs NoSQL, comparison, Cassandra, MongoDB, Couchbase.*

1 Introduction

Interactive software (wherein a user can offer input and receive output in real time) has changed fundamentally throughout the last 40 years. The online systems of the ’70 have evolved through a series of intermediary stages, into the "web-" and mobile applications we see today. These systems solve new problems for potential user populations that are far larger and they are being executed using a computational infrastructure that has suffered major changes especially throughout the last few years.

The architecture of these software systems has transformed as well. A modern web application can support millions of users simultaneously by spreading the load into a collection of application servers, managed by a load distribution system. Modifying the behavior of applications can be done progressively, without first creating “downtime” periods, by progressively updating the software on the individual servers that make up the system. Adjusting the load capacity of applications is easy to do by changing the number of available application servers.

In spite of this, the database technologies being used have mostly failed to keep up. The technology of relational databases, invented in 1970 is still widely in use today, even though it has only been optimized for the user types and infrastructure of that era. While a number of hacks and additions (e.g. distributed caching and data denormalization) have extended the life of these technologies, these tactics eliminate the key benefits of the relational model, and contribute to the growing complexity and expansion of the system.

Google, Amazon, Facebook and LinkedIn have been among the first companies to discover the serious limitations of the technologies behind relational databases as far as the demands of newer applications are concerned. Because commercial alternatives did not yet exist, they invented new approaches to manage their data. Their pioneer work generated a major interest, because an ever increasing number of companies was facing similar challenges. Within a short time-period open-source database projects emerged, to which the big companies flocked.

The premises to developing NoSQL: Big Data, Big Users and Cloud Computing

***Big Users*.** Not long ago, 1000 users/day was a lot for many applications and 10.000 represented an extreme case. Nowadays, most applications are cloud based and are available all over the internet, so they need to be able to accommodate users 24h/day, 365 days/year. Worldwide, more than 2 billion people are connected to the internet and the time they spend online is rising day by day, creating an explosion in the number of concurrent users. At this time it is not unusual for an application to have millions of users in one day.

***Big Data*.** Data has become easier to collect and access through intermediaries like Facebook, D&B and others. Personal data, spatial data, user generated content, log-in details are just a few examples. It is no surprise that developers place more weight on using this data both for improving the current applications as well as improving new ones.

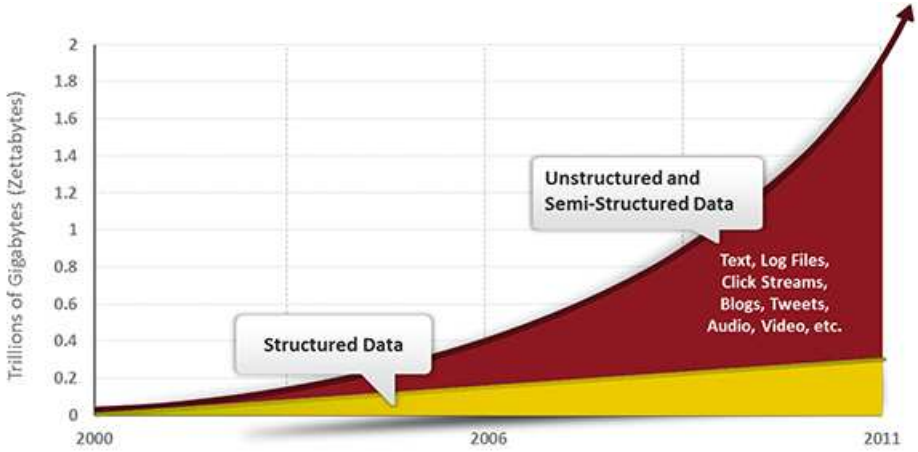


Fig. 1. Big Data: About 80% of the data generated now is unstructured or semi-structured. The total amount of data is growing very fast.[14]

***Cloud Computing*.** Not long ago, most consumers and many business applications were single-user applications which worked on a personal computer. Applications with a large volume of data, multiple users used a 2-level client-server application which ran behind the firewall and allowed a limited number of users. Nowadays, most new applications (be they consumer or enterprise grade) use an internet architecture with 3 layers, run on a public or private cloud and allow for a higher number of users. With this change of application architecture, new business models such as software as a service and advertising based models have become more wide-spread. [1].

These three aspects, highlighted earlier have led to the inevitable adoption of a different database technology which should keep up with the dynamics of interactive applications

2. Shifting from relational to NoSQL – a brief comparison

Scaling model - Relational databases are a technology that scales vertically – to add capacity (data storage or I/O capacity) we need a bigger server. The modern approach to application architecture is to scale horizontally, rather than vertically [2].

Instead of buying a bigger server, we use many commodity servers, virtual machines or cloud instances behind a load balancer. In reverse, system capacity can be easily reduced when no longer needed. While scaling horizontally is already common at the application logic tier, the database tier is just starting to use this approach.

Data model - The deployment benefits of NoSQL technology for scaling horizontally frequently get the most attention, but equally important are the benefits granted by a schema-less approach to data

management. With a relational database, we must define a schema before adding records to the database. Each record added to the database must strictly comply to this schema and its fixed column names and data types [2]. Bringing changes to the database schema is difficult, especially when it is a partitioned relational database that spreads across multiple servers.

If our data capture and management requirements are constantly evolving, a rigid schema quickly becomes an obstacle to change. NoSQL databases (whether it is a key-value implementation, document-oriented, column-oriented or otherwise) scale horizontally, and they don't require schema definition before inserting data nor changing the schema when data collection and management needs evolve [9].

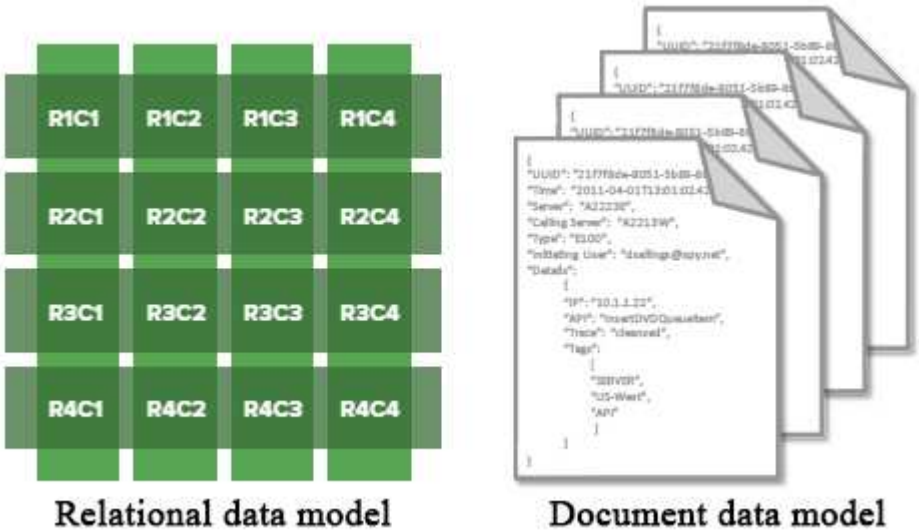


Fig. 2. Comparition relational - NoSQL

Relational data model – Besides the need to review the schema every time the data that we want to collect change, this model is characterized by the database normalization process, by which large tables are decomposed into smaller tables linked together. See the figure below:

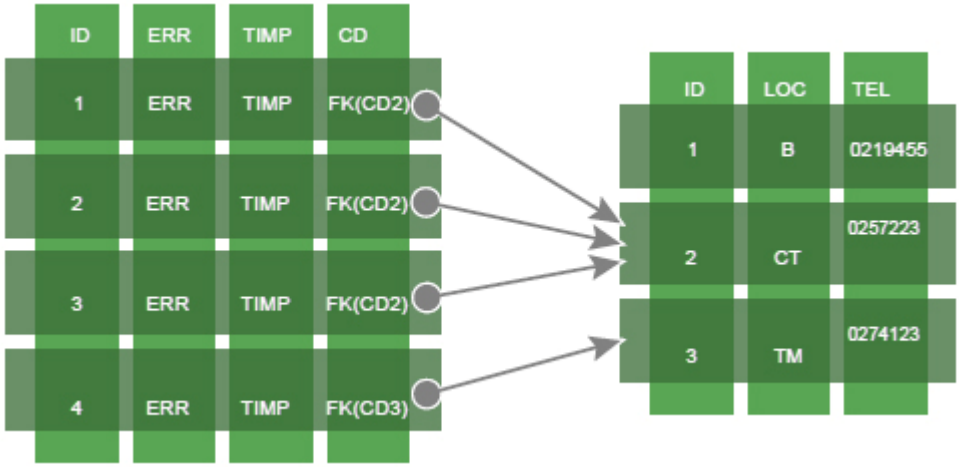


Fig. 3. Relational data model

In the above example, the database is used to store information into an error log. Each error record (row in Table 1) is composed of an error code (ERR), the time it took place (TIMP) and the datacenter (CD) in which it happened. Instead of repeating all

the information about the datacenter (phone, location), each error record will point to one row in the Data Centers Table (Table 2) containing the location of the datacenter (LOC) and the telephone number (NUM).

In the relational model, records are "spread" across multiple tables with certain attributes shared by several records (multiple error records contain the same data center information). The advantage is that there is less duplicated data in the database. The disadvantage is that a change to a single record can mean locking down multiple tables simultaneously, to ensure that the change doesn't leave the database in an inconsistent state. In a relational database, transactions can be complex, as the data, even of a single record, is spread about. This complex network of references between data items makes it very hard to distribute relational data across several servers and can lead to performance issues both when reading and when writing data.

Back when storage capacity was expensive and scarce, these compromises were justified. But in the last 40 years the prices of data storage units plummeted [11], and for many this compromise doesn't make sense any more. The use of more storage space in exchange for increased application performance and the ability to easily distribute the workload across multiple machines is now the best choice in many situations.

Regarding the **Document data model,** the use of the term "document" can cause some confusion. We need to clarify that such a database, has nothing to do with "documents" in the classic sense of the word. It doesn't mean articles, letters or books. Instead, in this case, a document refers to a data record that self-describes the data elements that it contains. Documents such as XML, HTML or JSON are examples of "documents" in this context. By using JSON [8] as the document format, the records of the error log shown earlier, would be:

```
{
"ID": 1,
"ERR": "Memorieinsuficienta",
"TIMP": "2014-03-16T23:59:58.75",
"CD": "BUC",
"TEL": "021.12.34.56"
}
{
"ID": 2,
"ERR": "Eroare ECC",
"TIMP": "2014-03-16T23:59:59.00",
"CD": "BUC",
"TEL": "021.12.34.56"
}
```

As we can see, the data is denormalized. Each record contains a complete set of information on the error without external reference. The records are self-contained. This makes it very easy to move the entire record to another server – all the information simply comes along with it. There is no concern that some parts of the record from other tables will be omitted. And because only the independent record (document) must to be updated when changes are made (instead of changing entries in many tables simultaneously), consistency at the record level is easier to accomplish [12]. Also data reading performance is increased.

However, complete denormalization of data is not required in a document-oriented database, as we will discuss in the next chapter. In fact, in the previous example, maintaining documents representing each datacenter and simply referencing those from each error record would probably be the correct decision. This separation would eliminate duplication and allow quick changes to information shared across multiple records (for example, if the phone number for the datacenter changed, there would be no need to go update all of its instances from the error log).

That said, data modeling decisions are dependent on the use case and future system changes.

Document-oriented modeling basics

Although it takes time for us to unlearn habits, by understanding alternatives we will be able to make more efficient use of your trusted knowledge as well. Finally, the instrument most suitable for a given task is the one that gives us the least headache. The more tools we know, wiser the choice we'll make.

Models

In an application, the data-containing objects are a central concept – being the model layer in the Model-View-Controller architecture (MVC) [13]. These are the documents that store our data and let us manipulate it. If a blog contains posts and comments, these will be implemented by two different models. Ideally, there should be a separate document for each post and each comment.

When we look at an existing application, we must stop at the Object-Relational Mapping (ORM) layer. Instead of dividing our models up into tables and rows, we transform them into JSON documents. Each document will receive a unique ID by which we'll be able to find later.

The primary Keys - in the world of NoSQL, the document ID is the one and only key of a document. These IDs can be seen as equivalent to a primary key in a relational database [9].

Some NoSQL database systems sort the documents by ID. Data with nearby IDs can be accessed more efficiently than IDs scattered in several places. Retention of data that is accessed at the same time, in the same place makes application faster.

Search by ID, being extremely fast, is the strength of this approach, and by using clever IDs we can ease our work very much. An example would be the use of prefixes to group our documents (user: component.example:xyz123).

Multiple occurrences and editability

Suppose we have a piece of data that shows up all over the application but we still want to be able to edit that data. For example, the title of a photo on flickr. The photo can show up in the photo stream, in sets, collections, groups on our flickr main page and in many other places.

Normally, a photo's title is shown with the photo. We could create a document for each instance of the photo in each of the locations. But then, if we change the title of the picture, we need to update a lot of documents.

If we know this is a finite number (no more than 10-100 e.g.) and the renaming doesn't have to happen at the same in all places (which means that an asynchronous background task could do the renaming), using separate documents for each instance can work fine.

However, in case the number of copies isn't finite and could potentially lead us to update thousands of documents that approach probably won't work. Instead, we would wish to store the title and perhaps other identifying data in a single "photo information" document and create a separate "photo placement" document for each location where the photo appears (these "photo placement" documents would each point to the photo's information document). Now when we display a photo we will make two lookups: one for the document containing the placement and another for the document containing the photo information. If we want to edit the title of a photo, we just edit the document containing the photo information and the changes will take effect everywhere on our site.

With the technique of "view collation", we can use a single query to return all the data we need. With views, we can keep a single canonical source for a sequence of data that is displayed in many different places.

In the world of relational database systems we are taught to normalize the data as much as we can; but in the NoSQL world we are taught to denormalize as much as possible. In both cases, the truth is somewhere in the middle.

Concurrent access

Getting back to the blog example. There are several authors, perhaps an editor, and each of them is working at a single article at any given time. Usually two people don't work on the same article. If we have data that we know is only edited by only one person at any given time, it's a good idea to store the data into a single document.

Comments on the other hand, are different. More people can write comments and they can do so simultaneously and independently. Once the post is published comments can be added immediately. To avoid write interlocking – in other words, concurrent writes happening to the same document – we can store comments in separate documents [10], ensuring that only one author is editing a single document at any given time.

To avoid serializing and locking each comment author out, or accidentally overwriting any data, just store the posts ID with the comment to be able to fetch them back in one request for displaying. (Note: document-oriented databases won't allow overwriting data, but will need more complex code to handle that case, so it's best to avoid this scenario, if possible.)

3. Cassandra, MongoDB and Couchbase – comparative aspects

The NoSQL databases have become a good alternative to BDR, especially for the applications that has to read and write quickly an enormous data quantity. They offer high efficiency, low response times, and horizontal scaling. In any case, with so many options available, choosing a NoSQL database can be complicated [3].

In what follows, we outline an overview of three of NoSQL databases on the market.

Cassandra is a distributed columnar key-value database that uses the eventual consistency model. Cassandra is optimized for write operations and has no central node: data can be read from or written to any node in a cluster. It provides a continuous horizontal scalability and has no single point of failure: if a node in a cluster fails, then another node comes and replaces it[4]. At this point, Cassandra is an Apache 2.0 licensed project, supported by the Apache community.

MongoDB is a NoSQL database, document-oriented, schema-free, which stores data in BSON format. A document based on a JSON, BSON is a binary format that allows quick and easy integration of data with certain types of applications. MongoDB also provides horizontal scalability and has no single point of failure. A MongoDB cluster is different from a Cassandra cluster or CouchBase cluster, because it includes an arbiter, a master node and multiple slave nodes [5]. Since 2009, MongoDB is an open source project with AGPL license held by the 10gen.company.

Couchbase is a NoSQL database, open source, document-oriented, designed for interactive web applications and mobile applications. Couchbase Server documents of are stored as JSON. With integrated caching, Couchbase offers low latency read and write operations, providing linearly scalable throughput. Architecture has no single point of failure. The cluster is easy to be scaled horizontally and live-cluster topology changes are supported. This means that there is no application downtime when updating the database, the software or the hardware using rolling upgrades. Couchbase Inc. develops and provides commercial support for the Couchbase open source project that is Apache 2.0 licensed.

Key criteria for choosing a NoSQL database

When choosing the right NoSQL database for interactive applications, these issues are key selection criteria that should be followed and analyzed [6]:

Scalability. It is difficult to predict when an application needs to scale, but when site traffic suddenly increases and the database does not have enough capacity, rapid scaling is needed, upon request and without changes in the application.

Similarly, when the system is idle, it should be possible to reduce the amount of resources used. Scaling the database should be a simple operation: we should not hit the complicated procedures or to make any change to the application.

Horizontal scalability of NoSQL database involves dividing the system into smaller structural components hosted on different physical machines (or machine groups) and / or increasing the number of servers that perform the same function in parallel. The following table summarizes the scalability aspects of the three databases analyzed.

a) Cassandra

Meets the requirements of a system with ideal horizontal scalability:

• The cluster automatically uses new resources;

• A node can be removed using an automatic or semi-automatic operation.

b) MongoDB

It has a number of functions related to scalability:

• Automatic sharding: auto-partitioning of data across servers;

• Read and write partitions: shards;

• Reads can be distributed over replicated servers;

• The cluster size can be reduced only by hand when the system is idle;

•The administrator uses the Administration Console to change the system configuration. Thereafter, the MongoDB server process can be stopped safely on the inactive machines.

c) Couchbase

It scales horizontally, too:

• All the nodes are identical and easy to install;

• Nodes can be added or removed from the cluster, with a single click and without changing the application;

• Sharding your data automatically evenly distributes data across all nodes in the cluster;

• Cross replication between data centers make it possible to scale a cluster from data centers for better data localization and faster access to them.

***Performance*.** Interactive applications require very little reading and writing latency. The database should provide consistently low latency regardless of task or data size. In general, reading and writing latency of NoSQL databases is very low, because the data is shared between all nodes in a cluster, while the application's working set is in memory.

***Availability*.** Interactive Web applications require a highly available database. If the application is offline, the business is losing money. To ensure high availability, the solution must do online upgrades to easily remove a node for maintenance, without affecting the availability of cluster to handle online operations, like as backups, and offer solutions for disaster recovery, if the entire datacenter crashes.

The paragraphs below show how *availability* is shaped in Couchbase, Cassandra and MongoDB:

a) Cassandra

• Each node in a cluster is given a data set that it is responsible for;

• If Cassandra has to process a write operation designated to be stored in a node that has failed, it will automatically redirect the write request to another node, which saves the write operation with a clue - a message that contains information about the node that failed;

• The node that holds the clue monitors the cluster to recover the failed node writing request. If the node is reconnected, the node holding the token will resend the message to it, so writing requests to be in their proper places;

• When a new node is added to the cluster, the workload is also distributed to it.

b) MongoDB

• Here, data is divided into several replica sets (shards);

• Usually, each of these consists of multiple Mongo Daemon instances, including an arbiter node, a master node and several slave nodes;

• If a secondary node fails, the master node automatically redistributes the workload on the remaining slave nodes. If the primary node fails, the referee chooses a new "master";
• If the arbiter node fails and there are no remaining instances in the shard, the shard is considered to be dead;
• Regarding master-slave replication, a replica set can span across multiple data centers, but writes can only go to a primary instance in a datacenter.

c) Couchbase

• It maintains multiple copies -up to three lines- for each document in a cluster;
• Each server is identical and serves active documents and replicated. Data are evenly distributed across all nodes and clients are aware of the topology;
• If a cluster node fails, Couchbase Server detects the failure and directs replica documents to other currently active nodes. As to reflect the new topology, cluster map is updated, and the application continues to run without interruption;
• When adding capacity, the data is automatically rebalanced, also without any interruption.

Ease of development. Relational databases require a rigid scheme and, if you want to change the application, you must change the database schema, too. Regarding this, all three NoSQL databases have the following advantages:
• Flexible schema - when you want to add new attributes to a document, you do not have to modify any of the existing structural elements. Old and new documents can coexist without further changes;
• Simple query language - because data in a NoSQL document is stored in a denormalized state, you can make queries and updates using put and get operations.

Performance. Interactive applications must support millions of simultaneous users and manage different workloads - read, write, or mixed. Below, we present the results of performance tests developed by Altoros Systems, Inc. for Couchbase, Cassandra and MongoDB. A scenario that simulates an interactive application was created, and with the aid of the Yahoo Cloud Serving Benchmark tool (YCSB- https://github.com/brianfrankcooper/YCSB /wiki) average latency at different levels of the system load was measured.
Performance results of this analysis can be easily replicated. To do this, the following configuration may be used. The YSCB tool with customized connectors for this test can be downloaded from Github.

NoSQL database test configurations

a) Cassandra 1.1.2

• Cassandra JVM settings:
1. MAX_HEAP_SIZE = 6 GB (dedicated memory for the Java heap).
2. HEAP_NEWSIZE = 400 MB (total memory for a new generation of objects).
• Settings for Cassandra:
1. RandomPartitioner using MD5 hashing to equally distribute the rows among the cluster.
2. Memtable with a size of 4 GB.

b) MongoDB

1. Four shards, each with a replica; each shard is made up of two nodes, one primary and one secondary.
2. Journaling disabled.
3. Every node set to run two Mongo Daemon processes and four Mongo Router processes.

c) Couchbase 2.0 Beta build 1723

1. Single replication option enabled
2. 12 GB RAM used for each node

Test Results

Figures 4 and 5 show average response time at different flow levels for reads, inserts and update operations, latency measured from client to server and back. The lower the latency values are, the better.

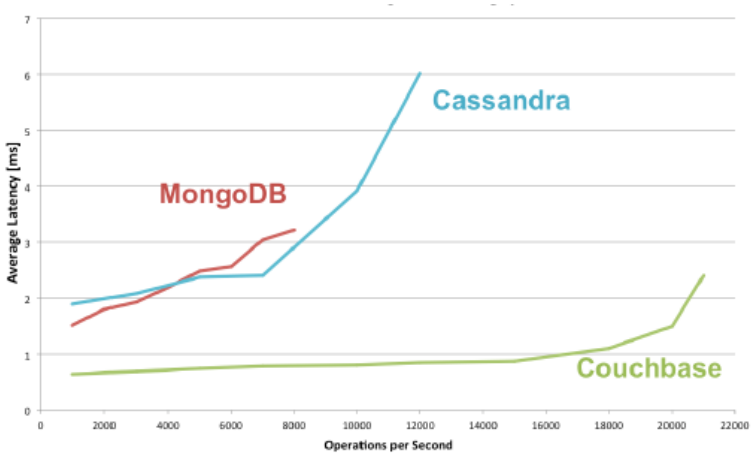


Fig. 4. Reads (medium latency) [6]

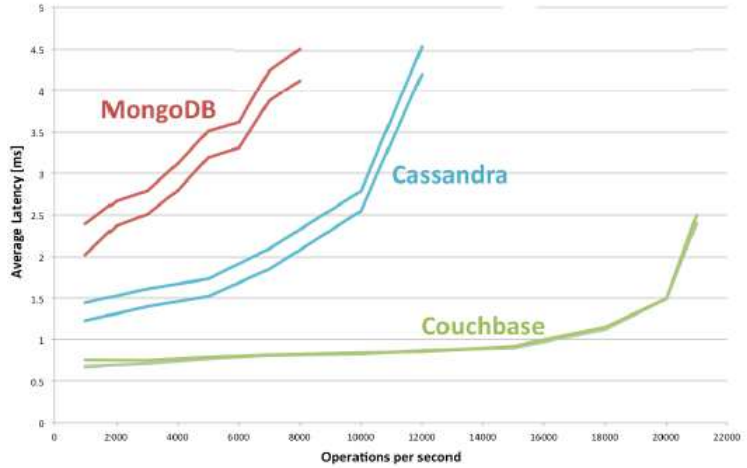


Fig. 5. Writes – insert, updates (medium latency) [6]

MongoDB processes the reading requests a little faster than Cassandra (Fig.4), but slower than Couchbase. Cassandra and Couchbase had better results at processing writes (Fig.5) compared with MongoDB. Cassandra uses cache key types and rows types, while MongoDB is based on memory-mapped files. Despite this difference, the two databases showed nearly equal read speeds (Fig.5). At writing, Cassandra had better results than MongoDB (Fig.5) because it firstly adds a data structure in memory, called Memtable. Then, if the configured threshold has been exceeded, it asynchronously sends data to the tables (SSTables) located on the disk.

4. Conclusions

Recent changes at the level of applications, users and infrastructure characteristics have determined application developers and system architects to seek alternatives to the relational data model, which is the standard for storing and retrieving data for more than 40 years. Many see technology in document-oriented database as a natural successor relational technology.

Regarding the performance of NoSQL databases analyzed Couchbase had the lowest latencies in the scenarios created for interactive applications because of cache objects built. Grained lock level document provides high efficiency for both types of operations, writing and reading.

Choosing a NoSQL database suitable for a particular application can be complicated because not all NoSQL solutions are the same. Each solution is optimized for different workloads and different use cases. Therefore, each has its own advantages and disadvantages.

However, the most important things to consider when working with large volumes of data are: latency, efficiency, availability, horizontal scaling and ease of development.

References

[1] Buyya, Rajkumar; Chee Shin Yeo, SrikumarVenugopal, *Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities*
[2] MongoDB, *NoSQL Databases Explained, (http://www.mongodb.com/nosql-explained)* accessed on March, 31st , 2014
[3] Altoros, *Using NoSQL databases for interactive applications (*www.slideshare.net/altoros/using-no-sql-databases-for-interactive-applications), accessed on March, 15th , 2014
[4] Datastax Documentation, *Architecture in brief. An overview of Cassandra's structure, (http://www.datastax.com/documentation/cassandra/2.0/cassandra/architecture/architectureIntro_c.html)* accessed on March, 25th, 2014
[5] Rick Grehan **,***MongoDB edges Couchbase Server with richer querying and indexing options, as well as superior ease of use*, InfoWorld, March 21st2013
[6] Alexey Diomin, Kirill Grigorchuk:*Benchmarking Couchbase Server for Interactive Applications*
[7] Brian Frank Cooper,*Yahoo! Cloud Serving Benchmark (*www.github.com/brianfrankcooper/YCSB/wiki) accessed on March, 30th , 2014
[8] ECMA International,*The JSON Data Interchange Format* (http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf) accessed on March,28th, 2014
[9] Highly Scalable Blog,*NoSQL Data Modeling Techniques* (http://highlyscalable.wordpress.com/2012/03/01/nosql-data-modeling-techniques)
[10] ShekharGulati, *How MongoDB Different Write Concern Values Affect Performance On A Single Node?* (http://whyjava.wordpress.com/2011/12/08/how-mongodb-different-write-concern-values-affect-performance-on-a-single-node/)
[11] Matthew Komorowski,*A history of storage cost (2014 update)* (http://www.mkomo.com/cost-per-gigabyte-update)
[12] Dare Obasanjo, *Building Scalable Databases: Denormalization, the NoSQL Movement and Digg*(http://www.25hoursaday.com/weblog/2009/09/10/BuildingScalableDatabasesDenormalizationTheNoSQLMovementAndDigg.aspx)
[13] Wikipedia, *Model–view–controller* (http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller)
[14] IDC, 2011 Digital Universe Study (http://www.emc.com/collateral/demos/microsites/emc-digital-universe-2014/index.htm)

Cristina BĂZĂR has graduated from the *Economic Cybernetics, Statistics and Informatics Faculty*at the *Academy of Economic Studies*, Bucharest (2012). At this moment, she is following the *Databases for Business Support* master program and is developping a disertation paper entitled*Optimizing the user experience within a web application regarding luminaries e-commerce*. Cristina's interests are broadly in the fields of databases, business intelligence, data warehouses, ETL and e-commerce.

Cosmin Sebastian IOSIF graduated in 2012 from the*Economic Cybernetics, Statistics and Informatics Faculty*at the *Academy of Economic Studies* inBucharest .At this moment he is pursuing the*Databases for Business Support*master program. His areas of interest are: Databases, Data Analytics, Geographical Information Systems and multimedia applications.

Business Intelligence. A Presentation of the Current Lead Solutions and a Comparative Analysis of the Main Providers

Bogdan-Andrei IONESCU, Sorina PODARU
University of Economic Studies, Bucharest, Romania
ionescu.bogdan.andrei@gmail.com, sorina.podaru@gmail.com

The aim of this paper is to synthesize the concepts behind Business Intelligence, by studying the solutions available on the market provided by the main players. We will present the software solutions already provided by them emphasizing the main advantages and benefits of each of them, but also as a comparative analysis, designed to reveal the area in which each provider is more remarkable than the others.
Keywords: *Business Intelligence, analysis, report, prognosis, management, OLAP, IBM, SAP, SAS, ORACLE.*

1 Business Intelligence

The concept of Business Intelligence includes, as stated in [1], a set of theories, methods, architectures and technologies, by means of which a great volume of primary data, most often historical data, is synthesized into information of interest such as key performance indicators for a company management. This information is presented in the form of reports, charts or tables.

Quality is the most important aspect when we talk about a correct implementation of any Business intelligence solution. This must be present in all of the four moments of the implementation: source data standardization, data processing, data warehouse implementation and reporting.

Regarding data warehousing, it has been noticed in time a change from relational to multidimensional. This change was necessary for the creation of dimensions as close to the user perspective as possible. Moreover, a multidimensional analysis offers a database much easier to consult and interrogate at a synthetic level, with less keys and administrative tables then in relational theory.

The reporting requirements of the companies have considerably changed during the last years becoming more and more complex in search of gaining the competitive edge.

The existing Business Intelligence solutions offer a diverse range of reports for company management, as described in [2], of which we distinguish:

- Static Reports. These reports are executed on demand or periodically, but always with the defining of the data prior to the execution.
  Examples: SAP Crystal Reports, Cognos Report Studio.
- Ad-hoc reports – Reports created by the end user, on demand, based on a standard template report.
  Examples: Cognos Analysis Studio.
- Interactive and multidimensional OLAP Reports – These reports offer the end user a high level of interaction by means of navigation techniques and data selection (Drill-down, roll-up).
  Examples: Oracle Discoverer, Cognos PowerPlay.
- Dashboard – Contains company aggregate, strategic and high-level data, in form of comparative and key performance indicators. They include both static and dynamic reports, charts and diagrams.
  Examples: Cognos Connection Portal, Oracle Hyperion, Microsoft SharePoint Server.
- Write-back reports – These interactive

reports are directly linked to the data warehouse, allowing data alteration. Most of the time, write-back reports are used for modifying and customizing products and clients' categories, for defining scenarios and forecasts, or for setting target volumes of sales.
Examples: Cognos Planning, SAP, Microsoft Access and Excel.

Gartner Group and Forrester Research are two of the biggest consulting and research companies in IT. Periodically, they analyze the providers of Business Intelligence solutions based on relevant criteria of evaluation, associating them with a type of provider within their Magic Quadrant [3] (Gartner – Magic quadrants squares: leaders, visionaries, challengers, niche players) or by assigning them scorecards (Forrester). Consistently in their analysis, between the leaders of the Business Intelligence technology, we find the four suppliers that we are going to present in this paper: IBM Cognos, SAP, SAS and Oracle.

2. IBM Cognos

"Cognos Company, part of the IBM group, is considered a global leader on the Business Intelligence market solutions in terms of performance management (BPM – Business Process Management). The company provides world-class software and services for activity planning and company strategy. The IBM solutions of Business Intelligence help companies plan, understand and handle the financial and operational results. Cognos Company was bought by IBM in February 2008." [4]
The main advantages are:

- The solution uses a powerful engine of analysis for dynamical planning and forecasting, a capability very appreciated by decision makers;
- The product allows the alignment of financial and operational objectives in one solution with a very short response time;
- The product has the ability to reduce the planning cycle by 75% and the report time from several full days to just a few minutes;
- It offers excellent performance for models and large data sets;
- IBM Cognos offers a dynamical and collaborative solution for sustaining the company's activities in all analysis and planning stages.

Many finance departments modify their strategy of support, performance management and risk management activities because the data validation process takes too long. These departments rely on individual solutions, which are entirely maintained by the IT department and which lack the performance and vital benefits to perform analysis activities on demand.
IBM Cognos radically transforms the planning, budgeting and forecasting processes, by executing "What-if" scenarios analysis, which facilitates the decisions and increases their quality.
Cognos TM1 is an analytic solution of planning which offers the following functionalities, as described in [5]:

- It allows an on demand analysis, planning, predictions and reports, and offers a complete view of the company's performance, including from the profitability point of view;
- It offers a various range of user interfaces, including Microsoft Excel, Apple iPad and web in order to cover a large range of work styles and personalized analysis on different types of activities;
- An easy browsing of large datasets, which uses exploring points to filter results. This product offers users the ever-necessary possibility of solving individual or work groups' problems. It is not a mere instrument of identification because it allows the transformation of information into actions in order to solve problems or to rapidly benefit from the rising opportunities, all

these functions being at hand on the desktop;

- It includes an innovative environment guided by models, which facilitate the development and implementation of planning solutions, analysis and forecast as the company necessities develop.

The features and benefits of Cognos TM1 include:

- A platform and a design that offer fast and complete answers for the entire company;
- Fast development of flexible models, including profitability models;
- Extensions to include risk analysis, rating matrices and reports;
- "What-if" analysis which immediately is reflected in diagrams and reports;
- The control belongs to the finance department because the final users can easily use this software without the need of technical knowledge.

This product offers a complete solution to organizations for all data types and for all components of the life cycle of a report: collaborative reporting, types of reports for the entire company, reports that are created once and can be used everywhere (Figure 1: Reporting in IBM Cognos).

Reporting features for this product are:

- Using a flexible model to create the reports, fields that can be modified for each object dimension;
- A report can have many output format types: HTML, PDF and Excel;
- The existence of reports templates that contain object, queries and standard layouts (Figure 2: Reporting în IBM Cognos).

Fig. 1. Reporting in IBM Cognos

Fig. 2. Reporting in IBM Cognos

3. SAP BI

"Having over 232 000 clients in 188 countries, SAP is the world leader in integrated software solutions for company management offering products for all domains. Likewise, best solutions are supplied for specific business processes, company resource planning solutions (ERP) which increase the productiveness of the entire company and Business Intelligence solutions which provide a rapid return of investment." [6]

The main advantages are [7]:

- SAP BI uses an intuitive dashboard, which allows a manager to have a real time overview of the planned demand and real demand of the products, organized by lead sources and geographical zone;
- Text analysis, coming from the social media such as: Facebook, Twitter and other sites and Web blogs, allows the company to redefine the marketing campaigns having a better understanding of people's opinion on its products;
- SAP BusinessObjects BI solutions redefine the way information is used by the company and offer new perspectives through a reliable Business Intelligence platform;
- The software helps organizations monitor data coming from various sources using a complex engine of event processing, by identifying the ones which can affect the day to day operations. The product offers information in real time by means of reports and notifications, enabling the users to rapidly react in these situations;
- An important benefit of this product is the fact that users can visualize a data quality rating, offering them a degree of reliability for each piece of information presented. Thus, a high quality is targeted for cases such as: compliance with the law. Other times high quality is not as important – for example, when a person wishes to find out people's perspective about a new product. In this case the decision remains to the end user, who will use this rating to make the best decisions.

SAP Business Intelligence is a product, which allows companies of any type or magnitude to visualize a diverse set of information in an intuitive graphic. With the SAP product a large quantity of information becomes a real strategic instrument, which helps organizations in their decision-making processes, based on a complex database analysis.

The solution was created with the purpose of increasing the quality of the decisions by implementing the following functionalities, as listed in [7]:

- The use of data warehouses for storing the report and analysis information and tools;
- The solution is addressed to all employees from all of the organization levels, helping them making fundamental decisions based on the provided information;
- SAP Business Intelligence assures the complete integration with all the other SAP components, which is very useful if one wants to extend the capabilities of Business Intelligence towards systems like CRM or ERP;
- This product is able to adapt to the changing demands of the contemporary society because it includes instruments through which the application content can be extended or integrated with other analysis and reporting instruments coming from third party providers;
- Another useful feature is the fact that the user can customize their interface and the access mode, as long as the security policies are followed.

The Business Intelligence solutions provided by SAP include data mining and predictive analysis algorithms, with the

data used for analysis exploring business scenarios from the past and present with the purpose of identifying the trends and outliers in order to predict future behavior.
SAP BusinessObjects Dashboard Software is a drag-and-drop instrument of visualization designed to interactively present strong dashboards, personalized with live connections to any data source. These dashboards can be shared with all business users involved in the decision-making process, so that they can easily agree upon the necessary measures.
This BI solution allows users to explore a large quantity of miscellaneous heterogeneous information in a very short time. Users can use common keywords words to find information stored in the data sources enabling them to explore the data in a direct way, without needing supplementary information regarding existing reports or data structure.
The functionality of reporting allows the exploring, visualization and supplying of reports using company applications or the web. This is ideal when questions are preset and each reply structure is known in advance. (Figure 3: Reporting in SAP).

Fig. 3. Reporting in SAP

4. SAS

"SAS is the leader in software and services for business logic analysis, and the largest independent vendor in the Business Intelligence market. Through innovative solutions delivered within an integrated framework, SAS helps customers from more than 60,000 locations to improve performance and deliver value by making faster and better decisions." [8]
The Business Intelligence solution from SAS provides complete end-to-end technology and also for data access and data quality. This ensures the fact that large volumes of raw data are processed into accurate reporting and analysis information.
For many users the most common tool of this product will be the SAS reports. These are dynamic views of a map of information, which can be displayed as a table, a chart, or a report (Figure 4: Reporting in SAS).
The main requirement of the companies is to perform an analysis of the problems, basic testing and the modeling of the results. SAS Enterprise BI Server product provides many tools for financial analysis, forecasting and statistical analysis, which is critical to solving problems and to allow the company to be more competitive.
The main advantages, as described in [9]

are:

- One of the main benefits which SAS Enterprise BI Server offers is that it allows the IT department to focus on other important tasks while still remaining part of the business by controlling the implementation, maintenance and security of BI platform. However, all employees can work on the platform for analysis and operational reports;
- Another advantage is that the product addresses a very broad area of fields, with the ability to store data and reports in one location. Also, an important benefit is that the product can work with other applications, such as Microsoft Office. Users can export their analysis and reports to Word, Excel, PowerPoint and Outlook;
- SAS Enterprise BI Server has a web portal that provides the functionality "point- and-click" which enables users to manipulate the layout and contents of the interface. It is also possible to transport data to the dashboard;
- Enterprise BI Server also has viewing functionality, which allows users to observe the analytical results that cannot be shown in a typical chart. Analytical results are displayed in a more interactive way, such as tables with bubbles, and even 3D videos containing data presentations;
- Another important feature is the Web-based reporting and distribution. Web-based reports have predefined layouts and custom templates and users can build and manage reports from multiple data sources. Reports can then be exported to PDF or Excel.

Fig. 4. Reporting in SAS

The features and benefits of SAS include:

- OLAP (On - Line Analytical Processing) refers to how business users can access the data using sophisticated tools, which allow navigation between different dimensions and hierarchies, such as "time" or "location";

- Query and reporting - the possibility of receiving answers to questions about the data that were not available before. This generates reports, which respond to these questions and can be saved for future viewing;
- Advanced Analysis - sometimes referred to as data mining or predictive analytics and forecasting, it uses statistical analysis techniques to predict trends;
- Collaborative BI - refers to the concept of taking data structured analysis and reports that have unstructured content (such as comments, discussions, accessories) so that data can be enhanced with additional content;
- Corporate Performance Management (portals, scorecards, dashboards) usually provides a container for several pieces that connect in such a way that the aggregate is presented as a story;
- BI in real time - enables real-time distribution of reports via e- mail, messaging systems and interactive displays.

5. Oracle BI

The appearance of Oracle Business Intelligence came naturally, due to the company's great history in database technologies and powerful software solutions.

Therefore, Oracle had a solid basis for creating both standard and interactive reports, which synthesize great volumes of data, meeting the user's needs.

An important aspect in which Oracle has a different approach from the other providers is the use of the E-LT architecture for data integration, rather than ETL. [10] This involves uploading data immediately after extraction, with the transformation being done in the destination database. This architecture removes the need of the middle-tier server used for ETL processing.

The main advantages of Oracle Business Intelligence, as described in [11], are:

- It eliminates redundant costs by standardizing a single platform for Business Intelligence;
- It provides a common infrastructure for creating reports, dashboards, ad-hoc analysis and OLAP analysis. The latter has a high degree of interactivity;
- It offers the users the opportunity to create their own reports and queries without the assistance of the IT department;
- Rapid highlighting of trends in very large data sets;
- Provides flexibility and scalability for the business solutions as the company implementing BI solution expands.

The features and benefits of Oracle BI include:

- The multidimensional analysis of data using the OLAP server enables users to create "what -if" scenarios used to predict business performance levels for different conditions;
- Scores and the management of the companies' strategies can be highlighted for each level of the organization and may be linked to key performance indicators so that performance can be measured, monitored and managed;
- Interactive dashboards are available in a Web interface for an instant view of metrics and important reports;
- The possibility of using a standard Business Intelligence solution, pre-configured by choosing the basic components;
- Users can quickly run their queries, and then analyze, visualize and share their results;
- Information is available through a variety of manners including: web-based dashboards, search bars,

ERP, CRM applications, mobile devices and Microsoft Office applications.

Oracle offers a wide range of software products for Business Intelligence. Oracle Business Intelligence Standard Edition One is a product for mid-sized companies in course of development. The most advanced features offered by it are the interactive dashboards. Oracle Essbase is the product that provides the ability to predict future behaviors by running "what - if" analysis. (Figure 5: Reporting in Oracle).

Fig. 5. Reporting in Oracle

6. Comparative Analysis

Table 1. Comparative Analysis

Criteria	**IBM Cognos**	**SAP Business Intelligence Solution**	**SAS Enterprise BI Server**	**Oracle BI**
Type of companies that the product addresses	Medium sized	All types of companies	Medium and large companies	Medium and large companies
Type of application	Divided into various tools: Analysis studio, Query Studio, Report Studio	Applications integrated into one package	Web application with portal access	Integrated application
Interface	Separately structured interfaces	Integrated dashboards	Web application with integrated dashboards	Web applications, ERP and CRM applications, mobile devices, Microsoft Office applications

Customization	Customizations can be made by the final user	Customizations can be made only by the IT staff	"Point-and-click" capability for the final users to manipulate the interface	Customizations can be made both by IT staff and final users
Navigation	Easy navigation between reports	Navigations between reports is time consuming	Easy navigation between reports	Easy navigation between menu items and reports
Intuitive	Yes	Yes	Yes	Yes
Tools used for analysis	Analysis studio and Query Studio	Desktop Intelligence-OLAP Intelligence	OLAP analysis capabilities	OLAP analytics, Mobile BI, Enterprise reporting
Integration with Microsoft Office	Yes	Yes	Yes	Yes

In terms of cost, there are three issues to be discussed: the cost of licenses, the cost of implementation and the maintenance cost. If the latter two do not show significant variations for the four suppliers, regarding licensing costs there are major differences. An IBM Cognos or SAS license can cost significantly more than an Oracle or SAP license. [12]

SAS shows the greatest benefits in the process of data loading. Firstly, the product occupies an important position in the market of ETL technology, which enables data mapping and cleaning methodologies. Moreover, SAS Data Access has a wide range of connectors for each type of database, storage container or file system. SAS Data Access connects to popular applications such as Siebel, PeopleSoft, Oracle, Baan and SAP.

As well as BusinessObjects and Cognos, SAS Metadata provides powerful metadata services that simplify not only the ETL mappings but also data formatting throughout the organization. These metadata services are crucial in making data easily and reliably identified to users who prefer ad-hoc analysis.

SAS can have a cost disadvantage compared to Oracle or IBM, but it compensates by reducing operational costs and risks. These benefits also include flexibility and an easy manner of changing the architecture of BI systems and data interoperability as well.

7. Conclusions

Medium companies from a wide range of industries are using IBM Cognos software to manage costs, make a profit and to encourage development. This integrated solution for planning and Business Intelligence helps companies with the specific capabilities that they need - reporting, analysis and planning. A feature of the product is that in can be extended as the company grows.

The product offered by SAP addresses all types of companies, all kind of managers and analysts who actively participate in the decision-making process, regardless of their technical knowledge. They will make decisions based on facts within the organization, relying on the business intelligence solution as they can easily access relevant information whenever they need in order to better understand how business works. By using such an instrument the decision makers will be able to act quickly and confidently to get the expected results.

The BI model proposed by SAS Institute focuses on analysis and reporting tools within SAS platform, which provides a complex processing of the data, information analysis and interactive reports. The application allows analysis based on a multitude of variables and dimensions, which can be set by the user himself.

Oracle offers a complete package of

Business Intelligence solutions in an integrated and unified architecture. Reports can include a high-level of interactivity and are available on a wide range of applications. Oracle's solutions meet the needs of all types of companies (from medium size to large ones) that may have problems that require complex analysis and reporting.

To conclude with, the Business Intelligence solutions belonging to four of the largest companies presented in this paper are solid solutions and the decision of implementing one of them is based on the organization's needs and business development strategies.

References

[1] Wikipedia.org - http://en.wikipedia.org/wiki/Business_intelligence
[2] ETL-Tools.Info http://etl-tools.info/en/bi/reporting.htm
[3] Gartner Group - http://www.gartner.com/technology/core/products/research/topics/businessIntelligence.jsp
[4] IBM - ftp://public.dhe.ibm.com/software/data/ro/cognos_reporting.pdf
[5] IBM Cognos TM1 - http://www.csiltd.co.uk/PDFS/BI/TM1%20PDF.pdf
[6] SAP - http://theredpoint.ro/produse/sap
[7] Red Point - http://theredpoint.ro/produse/sap/sap-pentru-companii/sap-bi
[8] SAS - http://www.sas.com/en_us/software/business-intelligence.html
[9] SAS - http://www.sas.com/offices/europe/romania/news/preleases/2011_06_15_forecasting.html
[10] Oracle Data Integrator and Oracle Warehouse Builder Statement of Direction http://www.oracle.com/us/products/middleware/data-integration/odi-statement-of-direction-1922235.pdf
[11] Oracle - http://www.oracle.com/us/solutions/business-analytics/business-intelligence/overview/index.html
[12] Gartner Group - https://www.gartner.com/doc/2538415/survey-analysis-customers-rate-bi

Bogdan-Andrei IONESCU graduated from the Faculty of Cybernetics, Statistics and Economic Informatics of the Bucharest University of Economic Studies in 2012. He is currently attending the courses of the Databases - Support for Business Master's program, expecting to graduate in 2014. At present, he is a mobile applications developer at a local company.

Sorina PODARU graduated from the Faculty of Cybernetics, Statistics and Economic Informatics of the Bucharest University of Economic Studies and is currently attending the second year of the Master's Degree in Databases - Support for business. At present she has the position of Junior Technology Consulting Analyst in a global consulting company.