Academic Journal of Economic Studies
Vol. 1, No.4, December 2015, pp. 119–130
ISSN 2393-4913, ISSN On-line 2457-5836

Centralized vs. Distributed Databases. Case Study

Nicoleta Magdalena Iacob[1], Mirela Liliana Moise[2]

[1]Faculty of Finance, Banking and Accountancy, Dimitrie Cantemir Christian University, Bucharest, Romania, E-mail: nicoleta.iacob_2007@yahoo.com
[2]195 Secondary School, Bucharest, Romania, E-mail: mirela_195@yahoo.com

Abstract *Currently, in information technology domain and implicit in databases domain can be noticed two apparently contradictory approaches: centralization and distribution respectively. Although both aim to produce some benefits, it is a known fact that for any advantage a price must be paid. In addition, in this paper we have presented a case study, e-learning portal performance optimization by using distributed databases technology. In the stage of development in which institutions have branches distributed over a wide geographic area, distributed database systems become more appropriate to use, because they offer a higher degree of flexibility and adaptability then centralized ones.*

Key words Distributed databases, advantages, disadvantages, case study, e-Learning

JEL Codes: C87

1. Introduction

In centralized databases all data are managed by a single DBMS and placed on a single node, only users being distributed in the network. For centralized databases, major benefits are determined by a good data integration that ensures data consistency and easy management of transactions in strict compliance with the ACID properties (Atomicity, Consistency, Isolation and Durability).
The disadvantages are especially high costs of communication and a very low reliability and availability because any error that blocks access to the database break all activity on the network. A distributed database system consists of a collection of local databases, geographically located in different points (nodes of a network of computers) and logically related by functional relations so that they can be viewed globally as a single database (Ozsu and Valduriez, 2011).
In distributed environments we face new problems that are not relevant in centralized environments, such as fragmentation and data replication. A data fragment constitutes some subset of the original database. A data replica constitutes some copy of the whole or part of the original database. The fragmentation and the replication can be combined: a relationship can be partitioned into several pieces and can have multiple replicas of each fragment (Silberschatz, Korth and Sudarshan, 2010).

For a database management system to be distributed, it should be fully compliant with the twelve rules introduced by C.J. Date in 1987 (Date, 1987): local autonomy; the absence of a dependency from a central location; continuous operation; location independent; fragmentation independent; replication independent; distributed query processing; distributed transaction management; hardware independent; operating system independent; independent of communication infrastructure; independent of database management system.
From the perspective of distributed databases, as effect of decentralization, data integrity, minimum redundancy and ACID properties should be relaxed because are very hard to accomplish, but increased availability is a major advantage.
The decision to use one or other of the solutions can be taken only after a careful analysis of application requirements, the size of the database, characteristics of available infrastructure together with evaluation of the global system performance.

1.1. Advantages of Distributed Database Systems

A major advantage of using a distributed database is that by sharing a database across multiple nodes can obtain a storage space extension and also can benefit from multiple processing resources. Although computational power has greatly increased in recent years, large data processing can lead to overall poor performance. By distributing data over multiple processing centers can obtain major performance advantages due to parallel data processing across multiple nodes, but keeping transactions ACID properties is much harder to achieve. In addition, distributed database systems offer other additional advantages:

- reflects the organizational structure of many organizations, given the fact that many companies are “distributed” geographically;
- increased reliability and availability. A distributed database system is robust to failure to some extent. Hence, it is reliable when compared to a centralized database system;
- local control. The data is distributed in such a way that every portion of it is local to some sites (servers). The site in which the portion of data is stored is the owner of the data;
- modular growth (resilient). Growth is easier. We do not need to interrupt any of the functioning sites to introduce (add) a new site. Hence, the expansion of the whole system is easier. Removal of a site also does not cause much problems;
- lower communication costs (more economical). Data are distributed in such a way that they are available near to the location where they are needed more. This reduces the communication cost much more compared to a centralized system;
- faster response. Most of the data are local and in close proximity to where they are needed. Hence, the requests can be answered quickly compared to a centralized system;

- secured management of distributed data. Various transparencies like network transparency, fragmentation transparency, and replication transparency are implemented to hide the actual implementation details of the whole distributed system.
- robust. The system is continued to work in case of failures. For example, replicated distributed database performs in spite of failure of other sites;
- compliant with ACID properties. Distributed transactions demands Atomicity, Consistency, Isolation, and Reliability;
- improved performance and parallelism in executing transactions can be achieved.

1.2. Disadvantages of Distributed Database Systems

The main disadvantages are:
- complex software – complex implementation. Costs more in terms of software cost compared to a centralized system. Additional software might be needed in most of the cases compared to a centralized system;
- increased processing overhead – It costs many messages to be shared between sites to complete a distributed transaction;
- data integrity – Data integrity becomes complex. Too much network resources may be used;
- different data formats might be used – This may cost time;
- deadlock is difficult to handle compared to a centralized system;
- may cause much more network traffic in case of write operation in a replicated form of distributed database;
- the data shared between sites over networks are vulnerable to attack. Hence, network security protocols must be used;
- more complex in terms of database design – according to various applications, we may need to fragment a database, or replicate a database or both;
- handling failures is a difficult task. In some cases, we may not distinguish between site failure, network and link failure.

2. Literature review

In this paper are presented theories regarding the actual stage of researches regarding distributed databases in order to improve them.
One classification is made regarding the data distribution mode: centralized or decentralized. In the case of centralized methods (Copeland *et al*., 1988; Didriksen *et al*., 1995; Hua and Lee, 1990; Ivanova *et al*., 2008; Tamhankar and Ram, 1998) there is a central site that obtain data and take decisions on fragmentation, data allocation or replication, while in the case of decentralized methods (Hauglid *et al*., 2010; Bonvin *et al*., 2010; Hara and Madria, 2006; Mondal *et al*., 2006; Sidell *et al*.,

1996; Wolfson and Jajodia, 1992) the decisions are taken by every site. There are methods that use a light decentralization scheme where sites are organized in groups and every group has a coordinator that is responsible for the decisions for the entire group (Hara and Madria, 2006; Mondal *et al*., 2006). In the model develped in phd thesis with title "Distributed Databases. A proposed dynamic model fully automated and decentralized", similar to DYFRAM the decisions of fragmentation, data allocation and replication are completely decentralized. Every site decide upon their fragments, and the decisions are made based on recent local writes and reads history (instead of connecting to queries processor and reading WHERE queries statements, the model is based on local access statistics).

3. Methodology of research

The optimal solution for the distribution can be defined by two *objectives*:

- *minimal cost*. The cost function consists of the cost of storing each F_i at a site S_j, the cost of querying F_i at site S_j, the cost of updating F_i at all sites where it is stored, and the cost of data communication;
- *maximal performance*. Performances are measured especially by system response times when dealing with read/write data processing operations.

Table 1. Comparison of data allocation strategies

	Centralized	Distributed		
		Fragmented	Complete replication	Partial replication
Storage costs	Small	Small	Large	Medium
Reliability and availability	Small	Small	Large	Medium
Update speed	Medium	Large	Small	Small
Communication costs	Large	Small	Large	Small
Redundancy	Small	Small	Large	Large
Concurrency access to data	Large	Large	Small	Small
Update time	Small	Small	Large	Large
Retrieval time	Large	Small	Small	Small

The total cost is calculated by adding the costs of communication (transmission of messages and related data), the costs of integrating and updating operations of each fragment F_i on each site S_j (CPU utilization and operations for input/output) and storage costs of each fragment F_i on each site S_j;

Response time of a transaction is calculated by summing the network transmission time and read/write operations processing time.
The allocation strategy involves the analysis of the requirements regarding data consistency, read/write operations percent in relation with system response time. Depending on these results, we can find the optimal combination between data distribution and data replication.

4. System monitoring

Enterprise Manager is a tool used for monitoring and administering an Oracle database. With Enterprise Manager organizations will benefit from increased productivity, ease-of-use, data integration, and dashboards tailored to their unique infrastructure. Using this tool is easy and effective to monitor, alert, manage and tune large numbers of database servers from an integrated, customizable graphical console. System monitoring features include monitoring functionality that supports detection and notification of a large area of IT problems.
From Figure 1 we can observe the active sessions and processor load on each session.

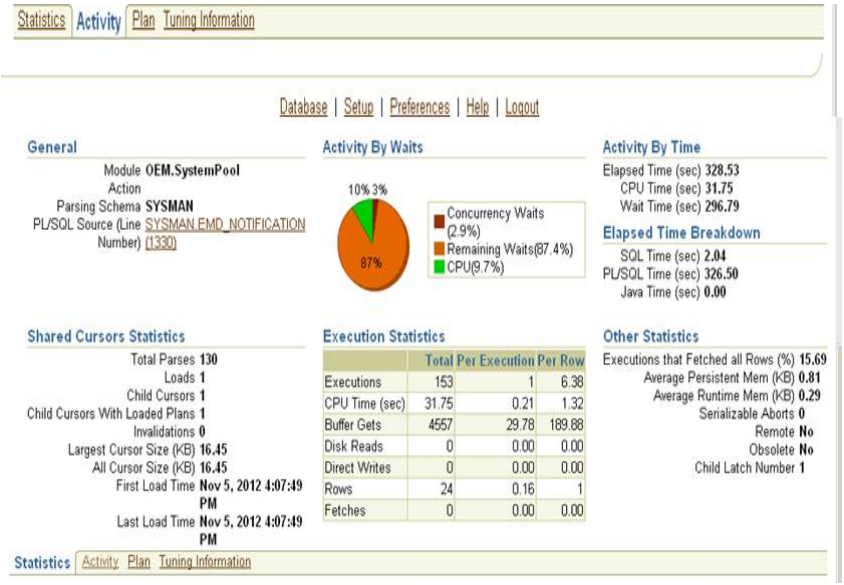

Figure 1. Database monitoring

Active sessions contain statistics regarding resources used by users connected to database.

5. Results. Implementing e-Learning Platform in an Academic Environment

The e-learning is a technology that has revolutionized the traditional system of distance learning (Defta, 2011), and its opportunities have been used not only by educational institutions, but also by different public or private organizations (Popescu and Bold, 2013; Popescu and Boroghina, 2015a, 2015b; Popescu and Radulescu, 2015). In terms of structure, an e-learning system provides facilities for the transfer of knowledge through the development and publication of educational content in the form of courses or virtual libraries, as well as testing of knowledge using simulations, scenarios or case studies for evaluation (Rădulescu and Rădulescu, 2011, 2012).
The users of the e-learning platform can access custom web pages (Pîrnău, 2009, 1010) depending on the group they belong, by which they can enable:

- administrative procedures by which each user defines its own activity context in training process;

View school situation for a student:

```
select u.prenume || ' ' || u.nume nume,
    (select fac.denumire from facultatib fac
      where fac.id =
          (select superior from facultatib fb where fb.id = fcurs.id)) facultate,
    fcurs.denumire materie,
    (select uprof.prenume || ' ' || uprof.nume from utilizatorib uprof
      where uprof.tipuser= 'P' and uprof.id =
          (select id_utlz from acces_facultatib
            where id_fact = doc.id_fact
          )) nume_profesor,
    doc.nota, doc.data data_sustinerii from utilizatorib u, facultatib fcurs,
documente doc
          where doc.tip = 'REZ' and doc.id_utlz = u.id and doc.id_fact = fcurs.id
```

Giving marks to students:

```
select u.nume,  u.prenume, f.denumire, d.* , decode(d.nota, null, 'Acorda nota',
d.nota) Notare
```

from documente d, utilizatorib u, facultatib f

where d.id_utlz = u.id and d.id_fact = f.id and ((upper(u.prenume||u.nume)= upper(v('USER')) and u.tipuser ='S') or (d.id_fact in (select f.id from acces_facultatib af, utilizatorib u, facultatib f where af.id_utlz = u.id and f.id = af.id_fact and upper(u.prenume||u.nume) = upper(v('USER')) and f.superior is not null))) and d.tip ='REZ'

- procedures for adding educational content to the platform (teaching materials, tests, video tutorials) (Figure 2);

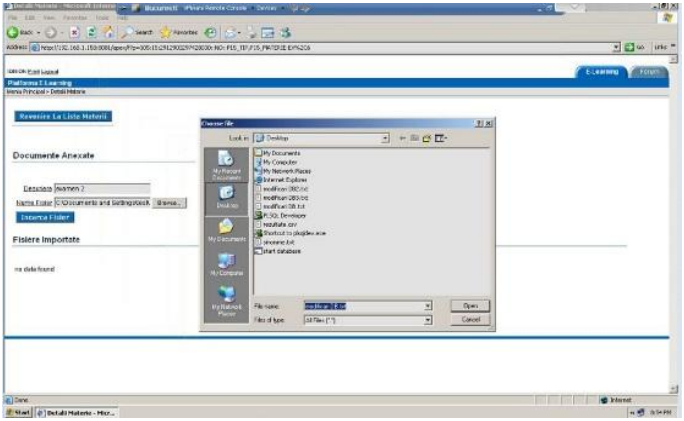

Figure 2. Loading documents

Materials list for a particular lesson <font color=blue style="font-size:18px;">&P1_SECTIA.</>

select f.denumire , f.id from facultatib f, acces_facultatib af, utilizatorib u where af.id_fact = f.superior and u.id = af.id_utlz and lower(u.prenume||u.nume)= lower(v('USER'))

Documents related to a lesson <font color=blue style="font-size:18px;">&P1_DENUMIRE_MATERIE_SELECTATA.</>

select td.denumire, td.id, td.tip,

(select count(d.id) from documente d , facultatib f

where d.id_fact = f.id and d.tip = td.tip and nvl(d.id_fact,:P1_MATERIE_SELECTATA) = :P1_MATERIE_SELECTATA) nr_doc from tipuri_documente td

• procedures for adding information regarding educational process (calendar of activities, advertisements, reminders of events) (Figure 3);

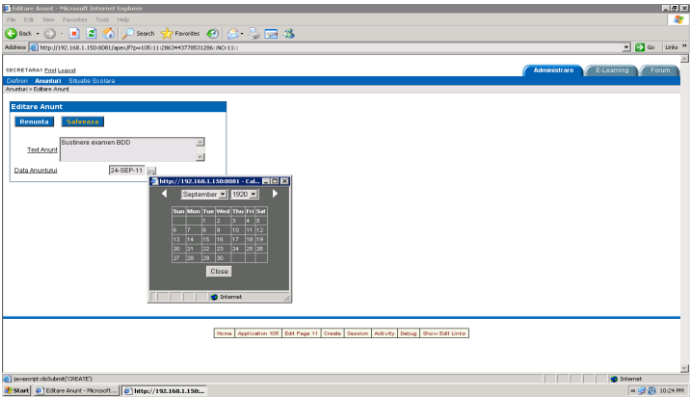

Figure 3. Adding advertisements

• procedures that support the educational process by synchronous contact (videoconferences) and asynchronous contact (discussion forums, e-mails) between teachers and students (Figure 4);

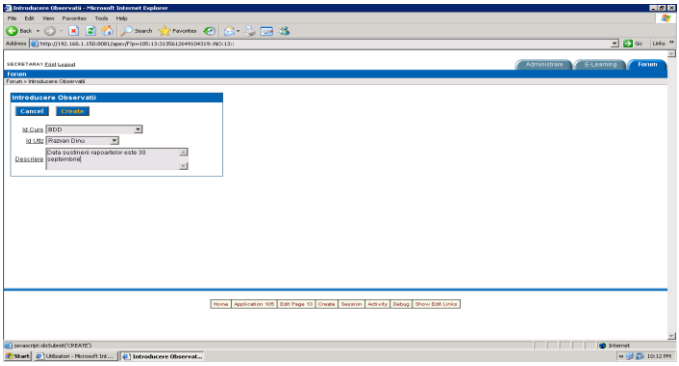

Figure 4 Posting advertisement on forum

Forum

select "ID_OBS", fo."ID_CURS", fo."ID_UTLZ", fo."DESCRIERE", f.denumire, u.prenume ||' '|| u.nume Utilizator from "#OWNER#"."FORUM_OBSERVATIIB" fo, facultatib f, utilizatorib u

where fo.id_curs = f.id and fo.id_utlz = u.id

- search procedures in virtual library (Figure 5);

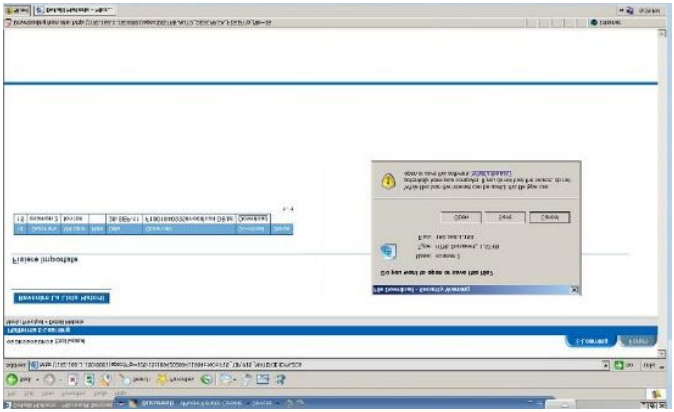

Figure 5. Downloading documents

- quality assurance procedures regarding the educational process (questionnaires, statistics) (Figure 6).

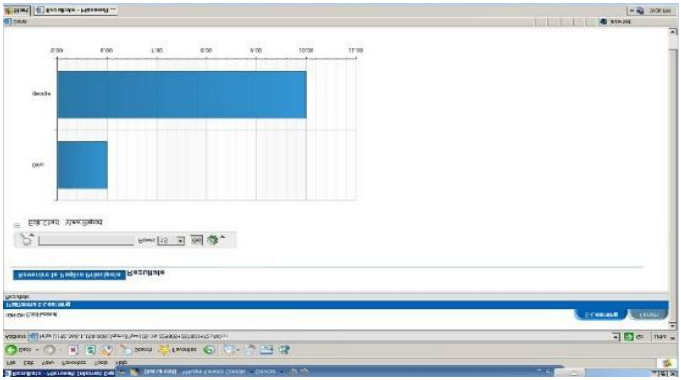

Figure 6. Viewing statistics

Thus, for each section of the portal there are some elements that must be fragmented or replicated between multiple sites (Iacob, 2011; Ciobanu and Ciobanu, 2012; Iacob and Defta, 2011). To better understand those concepts, we will give some specific examples.

The application's general objective is to provide logical support (software), in the procedural and functional plan, for the activities of an institution with geographically distributed locations using distributed databases. This is a significant performance optimization, because data are located on the site with "greatest demand" by using data fragmentation and assures a modular growth, because it's easy to add new systems in this configuration and also offer improved data availability by using data replication technique. The application consists of an e-learning portal built on Oracle Application Express that meets the growth and expansion challenges of a university with geographically distributed locations. The portal is based on three databases which are named intuitively, based on the city in which they are located: "Bucharest", "Timisoara" and "Paris".

To improve processing speed and data security it is necessary to partition database table which contains data about a university student, so that some records will be located in one site and others in another one. By this horizontal partitioning, the records of students who study in Paris will be stored in the database in Paris, since most requests of accessing table with students from Paris will be at the University of Paris. The disadvantage is that when an application needs to access all records of students in all towns within the university, it must collect data from each of these nodes. The second example is about the foreign languages courses, French for example, that must be found in the databases located in the countries where there is a branch of the university and where there are many requests for studying this foreign language, so there is a need for data replication. Distribution of these replicas has the objective to improve the speed of data processing operations and availability of transactions, so before we decide how to distribute the data we must determine the logical units of distribution.

6. Conclusions

Distributed databases eliminate disadvantages of centralized databases and offer several advantages such as availability, reliability, performance, modular development etc. But in distributed environments we face new problems that are not relevant in centralized environments, such as fragmentation and data replication. The use of distributed databases in e-Learning systems improves access to information and offer rapid data collection. Modern universities, with geographically distributed locations, must assume the responsibility to introduce new technologies in the educational process and must adopt learning methods based on the new technologies that are more efficient than the traditional ones. In other words, in order to be competitive, any university must take these technologies into account.

References

Bonvin, N.,Papaioannou, T.G. and Aberer, K. (2010). A self-organized, fault-tolerant and scalable replication scheme for cloud storage, *In Proceedings of SoCC '10*.

Ciobanu (Iacob), N.M. and Ciobanu (Defta), C.L. (2012). Synchronous Partial Replication – Case Study: Implementing e-Learning Platform in an Academic Environment, *Procedia - Social and Behavioral Science Journal*, Vol. 46, 2012, pp. 1522–1526.

Ciobanu (Iacob), N.M. (2014). *Distributed Databases. A proposed dynamic model fully automated and decentralized*, Ed. Pro Universitaria, Bucharest, 2014.

Copeland, G. et al. (1988). Data placement in Bubba, *In Proceedings of SIGMOD 1988.*

Date, C. (1987). *An Introduction to Database Systems*. Vols. I and II. 4th ed. Reading, MA: Addison-Wesley Publishing Co.

Defta (Ciobanu), C.L. (2011). Security issues in e-learning platforms, *World Journal on Educational Technology*, Vol 3, issue 3, 2011, pp. 153-167.

Didriksen, T., Galindo-Legaria, C.A. and Dahle, E.(1995). Database decentralization - A practical approach, *In Proceedings of VLDB 1995*.

Hara, T. and Madria, S.K. (2006), Data replication for improving data accessibility in ad hoc networks, *IEEE Transactions on Mobile Computing*, 5(11):1515–1532.

Hauglid, J.O., Norvald, H. Ryeng, and Nørvåg, K. (2010). DYFRAM: dynamic fragmentation and replica management in distributed database systems, *Journal Distributed and Parallel Databases*, Volume 28 Issue 2-3, December 2010, 157-185.

Hua, K.A. and Lee, C. (1990). An adaptive data placement scheme for parallel database computer systems, *In Proceedings of VLDB 1990*.

Iacob (Ciobanu), N.M. and Defta, C.L. (2011). The Impact of Distributed Databases in e-learning Systems, *Knowledge Horizons. Economics*, Vol. 3, No. 3–4, 2011, pp. 79-83.

Iacob (Ciobanu), N.M. (2011). The Replication Technology in e-learning Systems, *Procedia - Social and Behavioral Science Journal*, Vol. 28, 2011, pp. 231-235.

Ivanova, M., Kersten, M. L. and Nes, N. (2008). Adaptive segmentation for scientific databases, *In Proceedings of ICDE 2008*.

Mondal, A., Madria, S. K. and Kitsuregawa, M. (2006). CADRE: A collaborative replica allocation and deallocation approach for mobile-p2p networks, *In Proceedings of IDEAS 2006.*

Ozsu, M.T. and Valduriez, P. (2011). *Principles of Distributed Database Systems*. (3th ed.), New York: Springer, 2011.

Pîrnău, M. (2009). General information and main characteristics regarding Web Services protocol Soap and REST, *The Annals of University of Oradea*, vol. IV, 2009, pp. 1021-1024.

Pîrnău, M. (2010). Implementing Web Services Using Java Technology, *International Journal of Computers, Communications and Control*, 5(2), 2010, pp. 251-260.
Popescu, D.A. and Bold, N. (2013). Web application presentation of timetable for a university website, *The 8th International Conference on Virtual Learning*, October 25 – 26, Models and Methodologies, Technologies, Software Solutions, Bucharest, Romania, Ed. Univ. Bucuresti, pp. 253 – 256, 2013.
Popescu, D.A. and Boroghina, G. (2015). Students' computer-based distribution in highschool based on options as an extension of SEI distribution program, *The 10th International Conference on Virtual Learning*, October 30-31, Models and Methodologies, Technologies, Software Solutions, Timisoara, Romania, pp. 329-334, 2015.
Popescu, D.A. and Boroghina, G. (2015). Web-Based Programming Model, 6th *International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO'15)*, May 27-29, Istanbul, Turkey, IEEE Xplorer, pp. 1-4, 2015.
Popescu, D.A. and Radulescu, D. (2015). Approximately Similarity Measurement of Web Sites, *22th International Conference on Neural Information Procession*, Nov. 09-12, Istanbul, Turkey, Springer Proceedings, LNCS, Part IV, pp. 624-630, 2015.
Rădulescu, D.M. and Rădulescu, V. (2011). Educating the consumer about his right to a healthy environment, *Procedia - Social and Behavioral Sciences*, Volume 15, 2011, pp. 466-470.
Rădulescu, D.M. and Rădulescu, V. (2012). Sustainable development in terms of interpreting the human right to a healthy environment, *The Romanian Economic Journal*, Volume XV no. 46 bis, December 2012, pp. 111-120.
Sidell, J., Aoki, P. M., Sah, A., Staelin, C., Stonebraker, M. and Yu, A. (1996). Data replication in Mariposa, *In Proceedings of ICDE 1996.*
Silberschatz, A., Korth H.F. and Sudarshan, S. (2010). *Database System Concepts. (6th ed.)*. McGraw-Hill, 2010.
Tamhankar, A.M. and Ram, S. (1998). Database fragmentation and allocation: an integrated methodology and case study, *In Proceedings of IEEE Transactions on Systems, Man, and Cybernetics*, Part A. 1998, pp. 288-305.
Wolfson, O. and Jajodia, S. (1992). Distributed algorithms for dynamic replication of data, *In Proceedings of PODS'92*, New York, NY, USA.
http://exploredatabase.blogspot.ro/2014/08/advantages-and-disadvantages-of-distributed-databases.html, accessed on 15th February 2016.