Advances in Economics, Business and Management Research, volume 78

11th International Scientific Conference "Economics, Management and Technology in Enterprises 2019" (EMT 2019)

Distributed Data Processing in the Economy

Jaroslav Kultan
Department of Applied Informatics
University of Economics
Bratislava, Slovakia
jkultan@gmail.com

Peter Schmidt
Department of Applied Informatics
University of Economics
Bratislava, Slovakia
peter.schmidt@euba.sk

Miraigul Mukhambetova
Department of Informatics
L.N.Gumilyov Eurasian National University
Astana, Kazakhstan
mukhambetovamj@gmail.com

Marek Jalč
Department of Applied Informatics
University of Economics
Bratislava, Slovakia
jalcmarek@gmail.com

***Abstract*—The aim of the paper is to demonstrate the possibilities of efficient use of distributed database systems in the creation of the information system of the company. Distributed database systems using private computer networks make it possible to increase the efficiency of the information system, especially in enterprises that have multiple remote locations. In addition, it is necessary not only to design a suitable way to create a distributed database system, but to analyze the possibilities of preparing experts who are able to use the systems.**

Keywords—distributed database systems, private networks, information system, IS efficiency

I. Introduction

One of the main lack of enterprise information system with multiple subsidiaries in different cities or even states is the delay or even loss of information when registering data into the database system. There is also a problem with the transmission of a large amount of data, which can be greatly delayed as a result of busy networking. A major issue may be the suspension of the Information System (IS) due to the loss of connectivity to the database server and thus the impossibility of writing new data to the system.

The effective work in modern enterprises depends on the ability of the employees to access information resources from different parts of the world and to transfer data through secure channels so precious data does not fall into the competitors hands. To solve the above problems, it is appropriate to create distributed information systems with adequate security protection at the time of data transfer. The second element of IS is the use of VPN (Virtual Private Network), that provides a high level of security protection. This technology is the cheapest and does not cause problems for application end-users. In this case, enterprises can merge geographically separated stores and make secure transfer of informations [1].

II. Short Description of Distributed Enterprise

The headquarters of the company with its management is located in Bratislava. Our company owns five stores in Bratislava, Trnava, Banská Bystrica, Prešov and Košice. The company has two production halls in Bratislava and Košice. Stores are part of the production halls. Ground floor is used for transformation process where advertising products are produced – billboards, banners, t-shirts, etc. 1st floor is used for administrative and computer design. Each store has a parking lot for clients.

Stores, that are part of the production hall can add new products to offer. Individual stores can trade them but only if the system is fully functional and consistent. If it is not, they would not know at the store if the products are still available in other stores (on other nodes) or not. In case of a node or communication line failure, temporary data inconsistency may occur, but all transactions are recorded and once the node or link is restored, all overdue transactions are executed and a consistent DB status is ensured.

Production halls are divided into five sections. The first section focuses on the distribution and measurement of the necessary paper and canvas. The second one focuses on paper cleaning, paper scrolling and then the production moves to the printing section where the necessary visuals are applied to the layers. Additional materials are added in the next section, such as racks, ropes, etc. In the last section, the order is completed and ready for the customer.

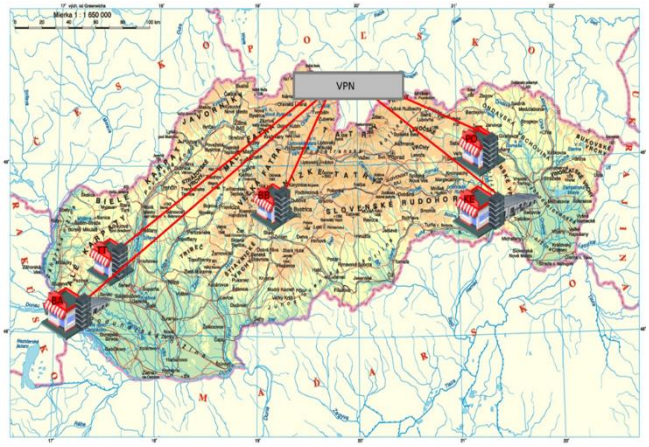

Fig. 1 Enterprise disposition. *(source: authors)*

In order to increase enterprise efficiency, improve data communication and data retention, we suggest creating IS using distributed data processing (DDP).

DDP is the transfer of computational resources to the user, the data and functions are scattered between the processing stations. For a manufacturing enterprise, it is important that each store has its own server that behaves as part of a single entity within multiple locations. Our information system is distributed in our company, territorially dispersed nodes are integrated so they appear to the user as a unit.

III. Distributed Database Systems

A new, modern way of using database systems involves the use of distributed systems with databases. This article lists not only methods for creating a database, but also methods of computing teaching on this kind of database.

Distributed database DDB is a logically unified database whose components are located in several network nodes. Users have one access to all the information stored in the database and the nodes can be geographically distributed [2]. The following functions of distributed databases can be distinguished:

- different nodes for receiving requests and multiple data nodes;
- Each node has its own operating system that works independently;
- data distribution mechanisms, such as horizontal and vertical fragmentation and replication;
- a hierarchical relationship in a distributed database (many clients - one server, many clients - many servers, equals) [3].

Each node uses its own database. Each DBMS of one distributed system node performs a complex multifunctional role and is one of the important stages of DDB development.

A. RDBMS(Relational DataBase Management System) Functions

Data Rules for Distributed DBMS, submitted in 1987:

1. Basic principle - From the end-user point of view, the distributed system should look exactly like unallocated one;
2. Local autonomy - nodes in the distributed system should be autonomous;
3. Deficiency of the reference central node - the system must not contain a single node which can cause system malfunction;
4. Non-continuous operation – there should never be a planned break in system operation to add or remove nodes or debris;
5. Independence of placement - the user must access the database from any node;
6. Independence from fragmentation - the user must access data regardless of the way of fragmentation;
7. Independence from replication (updates);
8. Distributed Requirements Handling- The system must support the processing of requests that refer to data located on more than one node;
9. Distributed Transaction Processing - The system should support the execution of transactions as recovery units;
10. Independence from device type - RDBMS should be able to work on devices with different computing platforms;
11. Independence from the operating system - RDBMS must be able to work on different operating systems;
12. Independence from Network Architecture - RDBMS must be able to work on networks with different architecture and media types;

B. Aspects of RBM Design

When creating distributed relational databases, the following design aspects arise:

1) Distribution. Each fragment is stored on the web, selected according to the "optimal" layout.

2) Replication. DDB can keep the current copy of the fragment in several different locations.

3) Fragmentation. With huge databases and a large number of nodes, it is advantageous to use fragmentation. Each relationship can be divided into several parts, called fragments, which are then distributed to different locations. There are two main types of fragments: horizontal and vertical. Horizontal fragments represent sub-groups of matrices of count n, and vertical fragments represent subsets of attributes. It should be noted that with a small number of nodes and a relatively small database, fragmentation is increasing the total system outgoings unnecessarily.

Design of DB should be based on both quantitative and qualitative indicators. Quantitative information is used as a basis for distribution, while qualitative information serves as a basis for creating a replication scheme. Quantitative information includes the following indicators:

1) the frequency of launching request for execution;

2) the place where the request is made;

3) Performance requirements for transactions and applications.

Qualitative information may include a list of transactions executed in the application, used relationships, access attributes of a new set, access type (reading or writing) or predicates used in read operations.

Definition of strategic objectives.

Local links. As far as possible, the data should be stored at nodes as close as possible to the place of use. Increased reliability and availability. The reliability and availability of data is enhanced by the use of the replication mechanism. In case of a single node failure, a copy stored on another node will always be available.

Acceptable performance level. Incorrect distribution of data will result in system barriers. In this case, some sites may be simply overload by the requirements of other sites, which can significantly reduce the performance of the entire system. Incorrect allocation can also lead to inefficient use of system resources.

Balance between capacity and cost of external memory. Be sure to consider the availability and cost of storage devices available on each system node. As far as possible, it is recommended to use some form of data replication on the nodes, such as mirroring or duplication of servers.

Minimize data transmission costs. In the case of DDB, the question is how far we can tolerate the inconsistency of data on individual copies of DB. You should carefully consider the cost of making remote queries in the system. If DB is consistent, data transmission overhead is minimal because only changes are transferred and not the whole DB. Naturally, when you increase the number of nodes, than the number of queries increases significantly. If the system latency reaches levels that are no longer acceptable, other solutions need to be taken, for example, mentioned fragmentation.

C. The role of data distribution

There are several alternative strategies for locating data in the system: centralized, separated (fragmented), full replication and selective replication.

Centralized placement. This strategy assumes creating a single database managed by system DBMS on one of the locations that will be accessible to all network users. In this case, the location of the links is minimal for all locations, except for the central one, because a network connection is required to obtain any access to the data. Therefore, the cost of data transmission will be high. The level of reliability and availability in the system is low, because a central location failure causes paralysis of the entire system.

Decentralized (fragmented) location. In this case, the database is divided into disjoint fragments, which every one of them is located in one of the spots in the system. If the item is located in spot, where it is most often used, availability will be high. In the absence of replication, the cost of storing data will be minimal, and the level of reliability will also be low. However, it will be higher than the previous version, because failure on any of these locations will result in loss of access to only that portion of the stored data. With a correctly chosen data distribution method, the performance level in the system will be relatively high and the cost of data transmission will be low.

Location with full replication. This strategy involves placing a full copy of the entire database on each system node. This will maximize link localization, reliability and availability of data as well as system performance levels. However, the cost of data storage devices and the cost of data transmission in this case will also be the highest.

Location with selective replication. This strategy is a combination of fragmentation, replication, and centralization methods. Some data fields are divided into fragments that allow high link localization, while others are used on many sites, but are not subject of frequent updates, are replicated. All other data are stored centrally. The purpose of this strategy is to combine all the benefits that exist in other models while eliminating their own mistakes. Thanks to its flexibility, this strategy is most commonly used [4] – [7].

IV. Description of Distributed Database System Using Virtual Private Network

In our study, a distributed database was developed and it is interacting with each node via VPN. In this study, each computer participated as a server and the data entered from one node were replicated to other servers through the specified network. Creating VPN tunnels helps protect data stored on the database server from unauthorized access.

At the beginning, we will look at the general organization of the Internet and we will show how nodes with servers can be located via VPN. In the general network structure, servers are designated as PS1, PS2, PS3, and PS4. Imagine that they are stores of companies located in different cities. Schematically, we could talk about VPN as a fully-fledged MESH (Fully connected network) topology.

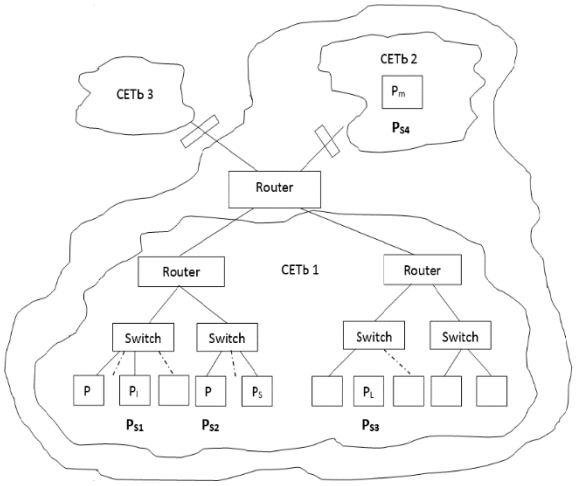

Fig. 2. Principle of a computer network schema. *(source: authors)*

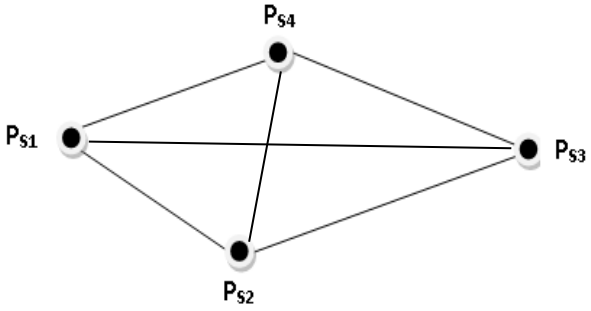

Fig. 3. Schematic topology of VPN. *(source: authors)*

There are lots of free applications that can create virtual private network, for example: TunnelBear, Airscribe, Hotspot Shield, Speedify, Private Tunnel, etc.

For our purpose, we used LogMeIn Hamachi software. The use of this software is very well suited for educational purposes, because it allows full interconnection of five nodes. This platform is also used by many companies in a commercial (paid) modification, because it provides intuitive control, reliability and necessary services.

In our study, we used free technology. The database was created in MySQL, where the Master - Slave implementation of data replication is already implemented. To write an application, PHP was used to control DB very easily. It is advantageous to use some of the WAMP, XAMPP, LAMP or MAMP distributions depending on the node´s OS. Each node runs a web server and a database server. You can access the app via a web browser.

Creating a distributed DB has several issues. Creating a DB itself and placing it on nodes is not a problem. Creating a

web application to access DB is also easy. Transaction logging (which is actually replication) to other nodes already requires some OS security experience as well as knowledge of antivirus programs. When node is connected to a VPN and when OS firewall and antivirus firewall troubleshooting is done, permissions to access remote databases still need to be set up. Without the proper settings, you will not be able to write or perform any tasks. In order for nodes to see and accept each other, we used the virtual IP addresses assigned by LogMeIn Hamachi. These IP addresses are fixed and therefore they can be used for node identifiers in the application. With this constellation, there are no theoretical barriers to writing transactions to all nodes in this VPN.

The problem occurs when one of the nodes fails. Since each node is autonomous and can perform all operations related to node activity, there is likely to be a global data inconsistency within the DDBS. Nodes that can see each other will continue to exchange transactions and all will remember those transactions that have not been sent to an unavailable node. If the inaccessible node is reconnected, the other nodes will send it all the transactions that are going to be executed on that node and the DB on all nodes will be the same, consistent. Another case of the incident is when the connectivity of one of the nodes is interrupted. In this case, all nodes can perform local transactions or transactions between nodes that are visible. All transactions that relate to an unavailable node are recorded. An inaccessible but functional node can perform local transactions, but at the same time all transactions to be processed on other nodes must be written. Once the connection is resumed, the nodes will send each other undone transactions, the nodes run them and the consistency of the distributed database system is restored.

There are many methods and ways of processing such a replication mechanism, but they mostly require programming. Master - Slave replication is available on the mentioned platform, which is applicable to 2 nodes. If Master - Multislave replication was implemented, the data replication necessary for DDBS only needs to be set up and nothing would be programmed, which would contribute to the development of DDB.

V. CONCLUSION

Using VPNs with distributed databases has lots of advantages in creating efficient information systems. For the purpose of their efficient use, it is necessary to improve the training of experts. Many studies have been devoted to effective education [8]. In addition to basic education issues, it is also necessary to increase the training level of experts in management area from small or medium enterprises, so they will be able to formulate their tasks and objectives in order to implement distributed database systems [9]. With a goal to introduce effective learning and new elements into enterprise management, it is essential to create the conditions for education to be effective and not teared away from practical experience. Such a task can be carried out by an Internet portal that includes study materials, procedures and guides for training in management, VPN creation, and distributed databases. The above-mentioned literature [10] outlines the basic principles of such a proposal.

REFERENCES

[1] V. V. Sheyda. Using TCP and UDP protocols for secure transmission of information over SSL-VPN tunnels // Reports of Tomsk State University of Control Systems and Radioelectronics. - 2010. - №. 1-2 (21). - p. 225-230.

[2] N. Kulagin. The Model of Building a Distributed Database for Corporate Information Systems // Bulletin of Volgograd State Technical University. - 2012. - V. 4. - №. 13.- C. 127-129.

[3] Serik M., Mukhambetova M.Zh. "The use of client-server technology in the content of the education of the university" // Math. Between.Conf.Modern education. Problems and solutions Italy (Rome). Publication in the journal "International Journal of Experimental Education" No. 8 (Appendix). - Moscow: RAE, 2017. - p. 62-65.

[4] Foster I. et al. Grid services for distributed system integration //Computer. – 2002. – №. 6. – C. 37-46.

[5] Kulagin N. V. The model of building a distributed database for corporate information systems // News of the Volgograd State Technical University. - 2012. - V. 4. - №. 13.- C. 127-129.

[6] Development of distributed relational databases. https://life-prog.ru/2_34402_razrabotka-raspredelennih-relyatsionnih-baz-dannih.html [appeal date: 1/28/2019]

[7] Problems and features of distributed databases. https://all4study.ru/bd/problemy-i-osobennosti-raspredelennyx-baz-dannyx.html [appeal date: 12/14/2018]

[8] M. Glatz, and B. Mišota, "Active education of the employees of the small and medium-sized enterprises," In *Knowledge for Market Use 2016: Our Interconnected and Divided World. International Scientific Conference*, Olomouc, Czech Republic, 2016, pp. 104-114.

[9] J. Chajdiak, M. Glatz Durechova, and B. Mišota, "The selected measures of innovation," in *Innovation Management, Entrepreneurship and Corporate Sustainability. Proceedings of the 4th International Conference*, Prague, Czech Republic, 2016, pp. 85-95.

[10] M. Zajko, and B. Mišota, "Design of information architecture and navigation of internet portal on support of regional innovation development," in *International Scientific Conference on Marketing Identity - Digital Life*, Smolenice, Slovakia, 2015, pp. 400-413.