DISTRIBUTING AND REPLICATING DATA IN HOSPITAL INFORMATION SYSTEMS

Marek Wojciechowski
Institute of Computing Science
Poznan University of Technology
ul. Piotrowo 3a, 60-965 Poznan, Poland
Marek.Wojciechowski@cs.put.poznan.pl

Abstract

Distributing and replicating data are techniques used to improve performance and reliability of information systems. This paper presents advantages and disadvantages of distributed architectures that are important in case of hospital systems.

In the paper, several issues that have to be addressed when planning a distributed architecture for the hospital information system are discussed. The particular emphasis is laid on possible problems that might occur in case of partial failure of a distributed hospital information system.

1. Introduction

Centralized systems are in general easier to design, implement and manage than distributed systems. But in some cases, it might be desirable to distribute data over a computer network in order to improve performance and scalability of the system or to move the data closer to end users. In a distributed database system [1][2], some of the data might be replicated to guarantee access to critical data, reduce network traffic and shorten the response time of the system. Distributing data and replication are powerful techniques but they must be used carefully after a thorough analysis of a designed system. In this paper advantages and disadvantages of distributing and replicating data in hospital information systems are presented with special emphasis on some potential problems that might occur as a result of distribution or replication of the data.

2. Distributed vs. centralized database systems

In centralized database systems all the data is managed by one server. In case of failure of the server there is no access to institution's data and the whole information system is paralyzed. Fortunately, if the server was administered

properly there should be possible to restore the state of the server from the moment of failure. Even if the database management system does not guarantee this there is always an option of going back to a consistent state of the database from the most recent backup.

The issue of fault tolerance and recovery is much more complicated in case of distributed database systems. A distributed database can be defined as a collection of multiple, logically interrelated databases distributed over a computer network [1] or a set of databases stored on multiple computers that typically appears to applications as a single database [3]. In a distributed system a single server failure does not have to affect the whole information system. Let us assume a hospital whose all organization units have their own database servers managing their data. If one of the servers crashes only one organization unit will lose access to its data. Other organization units will suffer only partial loss of functionality due to inaccessibility of one remote server and its data. But still from a single organization unit's point of view the risk of the loss of all functionality is not less than in case of a centralized system with additional possibilities of partial failures. Moreover, if a company or institution relies on one server it will probably improve its fault tolerance by using such techniques as disk or server mirroring, which might be too expensive in case of a multi-server environment.

It is no surprise that recovery complicates in a distributed system. Restoring the state of a single server after its failure from the most recent backup might not be enough because data on the restored server may be inconsistent with other servers. A distributed database management system must offer options that guarantee that results of all committed transactions will be restored after a server failure. For instance, Oracle gives an option of archiving redo logs which together with regularly made backups enable the administrator to restore a transaction consistent state of a database from the moment of failure.

Centralized systems are easy to design, manage and maintain but in some cases distributed systems may be more suitable. There are two most important reasons for using distributed database systems. The first is a geographic distribution of an institution or organization for which the system is being designed. If an institution consists of many organization units which are not connected by a stable and fast Local Area Network (LAN), which might be true for some hospitals, it would be irresponsible and inefficient to base the institution's database system on one server. Such a solution would lead to long response times and in case of unstable network connection to possible problems with access to data. A much more reasonable solution would be assigning a database server to each organization unit to manage its data and organizing the system in such a way that the servers cooperate with each other to enable access to remote data.

The second reason why it might be advisable to distribute the data over several servers is a need of increasing the processing power of the system. Hospitals, like other large institutions, process huge volumes of data, which even a powerful server might not be able to deal with efficiently. For big organizations, companies, or hospitals data distribution might be the only way to improve performance of the information system.

Additionally, distributed database systems offer better scalability than centralized systems. Besides the possibility of upgrading a single server (vertical scalability) there is an option of adding a new server to the system (horizontal scalability). Another advantage of distributed database systems, which may be important in case of a hospital information system is site autonomy (each organization unit has its own server managing its data and may decide which data should be made accessible for other organization units).

The biggest disadvantage of a distributed system is a possibility of losing an access to remote data in case of a remote server or network failure. Consequences of such failures are minimized by means of replication.

3. Replication

In distributed database systems replication is used to guarantee access to remote data by providing their local copies. It might also shorten response times of access to such data in case of a system where network connections are slow. A database management system offering replication must guarantee that changes to one of the replicas will be propagated to all other replicas so that they are consistent with each other. This can be done synchronously (if a transaction modifies one of the replicas modifications must be propagated to other replicas in the same transaction) or asynchronously (a transaction modifying one of the replicas ends and the system guarantees that modifications will be propagated to other replicas later). Asynchronous replication in case of inaccessibility of one of the replicas might lead to temporal inconsistency, which must be taken into account when replication is used in such critical applications like hospital information systems. On the other hand, synchronous replication is difficult to be used in practice because it will not allow to modify a replicated object if any of its replicas is inaccessible.

There are various replication models. Propagation of changes (synchronous or asynchronous) can be possible only in one direction (one master replica and many so-called read-only snapshots [3]) or in many directions (changes are possible for all replicas). The second option is more powerful but requires more sophisticated mechanisms and can be more expensive (as in case of Oracle Server [3]).

Replicas can replicate the whole table or a subset of it. In Oracle Server a snapshot can be based on any SQL query but propagation of changes (refreshing a snapshot) is much more effective in case of simple queries (access to one table, projection and simple selection).

4. Potential problems connected with data distribution and replication in hospital information systems

Distribution and replication of data in a hospital information system may lead to a complicated implementation of the system and, what is more important, to some dangerous behavior of the system in case of failure.

The first problem is to assure that all organization units use the same catalog data (lists of diseases, medical tests, doctors, etc.). This can be done by means of

one master copy and a set of read-only snapshots or by multi-way replication. The first solution leads to remote updates in case of modifying a catalog from many units in a hospital. The second requires more expensive versions of database management systems. In addition, if modifications have to be applied to one selected master copy, they will be impossible to perform when the master copy is inaccessible.

Another problem is connected with patient's case history and results of medical tests. A patient can move from one ward to another. His case history would be distributed and should be replicated to guarantee an access to the whole of it. It would be desirable if one ward replicated from other organization units only data concerning its patients. Unfortunately there are no techniques of doing it efficiently. Selective replicas would have to be based on complex queries and therefore could not be refreshed incrementally. To allow fast incremental refreshing of replicas containing patients' case histories, all the data should be replicated on each server. Enabling modifications of patient's records generated in other organization units raises the same dilemma as in case of catalog data.

Probably the most dangerous problems can be caused by using asynchronous replication to propagate information. If results of medical tests are propagated by means of asynchronous replication, it is possible that an invalid result entered into the database by mistake will propagate but information about canceling it will not because of a network connection failure, which can lead to a wrong decision by a doctor.

5. Conclusions

Distributed database systems are suitable for large institutions, which consist of many organization units, and therefore seem to be a good solution for hospital information systems. Replication of some data is necessary in a distributed system to guarantee access to data from remote servers and improve performance. Replication is especially effective in case of data generated in one organization unit and only read by other units. Nevertheless, asynchronous replication, which seems to be the only practical solution, must be used thoroughly because it can hide a network or server failure preventing the access to the most up-to-date information.

References

[1] M. T. Özsu, P. Valduriez, Principles of Distributed Database Systems, Prentice-Hall, 1991.

[2] S. Ceri, G. Pelagatti, Distributed Databases Principles and Systems, McGraw–Hill Book Company, 1984.

[3] Oracle8 Server Documentation, Oracle Corporation, 1997.