Update of Database Application Using Replication

Ľubomír Semančík

Armed Forces Academy of Gen. M. R. Štefánik, Department of Informatics,
Demänová 393, 031 06 Liptovský Mikuláš, Slovak Republic
lubomir.semancik@aos.sk

Abstract. This paper describes possibilities of updating of a database application using replication in distributed databases. This update is possible for those database applications in which the application logic is implemented by stored procedures and user – defined functions. The necessary procedures and functions are modified on a specific database server and then they are sent to the other servers.

Keywords: update of database application, distributed database system, replication, database.

1 Introduction

Software of an information system includes three basic components [3, 5]:

- Presentation functions – provide interactions between users and database applications, represent a presentation of results of selected tasks to users and provide an interface to application control (provide an user interface);
- Application functions – provide application logic (represent data processing in the application – performing necessary computations with data obtained from the database);
- Data management.

Nowadays the client – server architecture is the most used architecture for information systems and database applications. The database servers of this architecture have a possibility to define:

- stored procedures – precompiled sequences of SQL commands stored in a database;
- user – defined functions.

Stored procedures and user – defined functions are an option of implementation of application logic (applications functions).

Whereas, replication in a distributed database allows to send the stored procedures and user – defined function between computer network nodes. Then, an update of some types of database applications can be realized by means of replication.

2 Logical models of client – server architecture

The definition of client – server architecture is based on distribution of processing between several processes, where at least one process is client that requests services from server [3].

For client – server systems the following subgroups have been defined (Fig. 1). These subgroups are based on the distribution of functions between the client and the server.

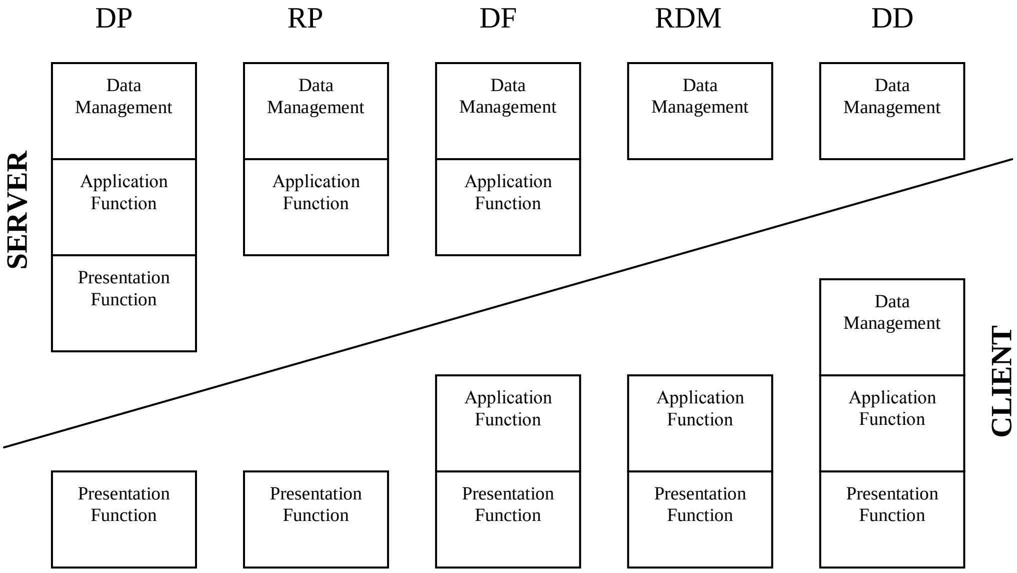

Fig. 1. Logical models of client – server architecture

Where:

DP – Distribution Presentation (Interface Distribution);
RP – Remote Presentation (Interface Separation);
DF – Distributed Application Function (Application Distribution);
RDM – Remote Data Management (Data Separation);
DD – Data Distribution.

3 Replication

A distributed database system is a set of interconnected computer network nodes, where each node contains a separate database system and the nodes can access the data stored on another nodes as if they were placed in their own node (Fig. 2) [2, 4].

Distributed database is thus a set of interconnected databases, which are located on different nodes so that the user is handled as in the case of a centralized database [2, 4].

The distributed data processing is linked with various forms of organization of data maintained by a distributed database management system. In principle, possible ways of distributing of data stored in the databases are the following [2, 3, 4]:

- Replication of data – all nodes contain the same data (the selected part of the same data).
- Fragmentation of data – the basic idea of fragmentation is a distribution of data on different servers.

A distributed database system can be divided into [2]:

- homogenous – each of the local databases is controlled by the same DataBase Management System;
- heterogeneous – each of the local databases is not controlled by the same DataBase Management System;

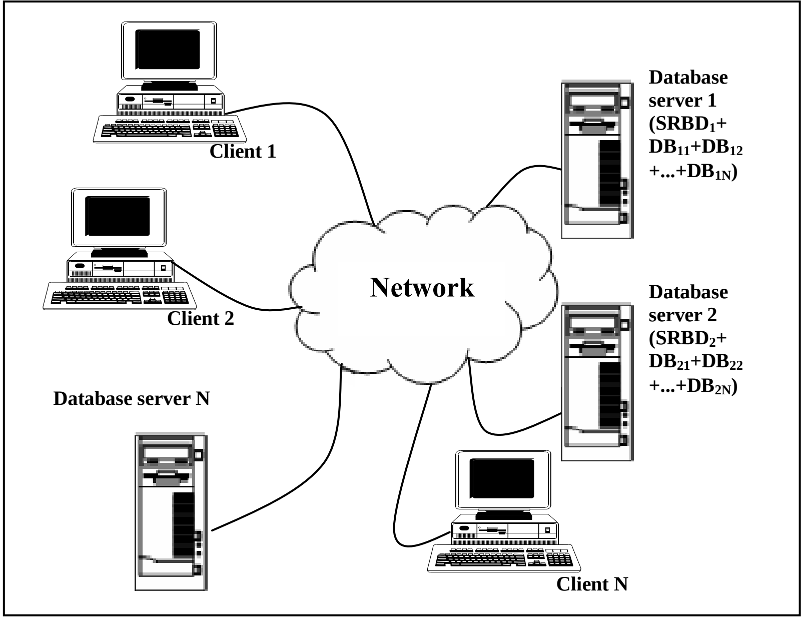
Client 1
Client 2
Network
Database server 1 ($SRBD_1$+ DB_{11}+DB_{12} +...+DB_{1N})
Database server 2 ($SRBD_2$+ DB_{21}+DB_{22} +...+DB_{2N})
Database server N
Client N

Fig. 2. A distributed database system

The basic idea of replication is to create a local database, which is identical with the central database (this database is redundant).

An importance of replication is [4]:

- A higher level of data accessibility;
- A higher level of data security.

The replication increases [2]:

- system performance;
- system availability.

The replication involves [1, 4]:

- Publisher
  - The database server, which provides a data and objects for replication (database tables, views, stored procedures, user – defined functions).
- Subscriber
  - The database server (one or more), which synchronizes a data, objects or transactions with publisher (subscribe data from publisher).
- Distributor (distribution database)
  - The database, which controls the replication.

Replication settings consist of [4]:

- Creating distribution database;
- Creating publications (selection of objects for replication);
- Setup of subscribers;
- Setup of the type of replication;
- Start of replication.

We know the following replication types [1, 4]:

- Snapshot replication – represents the current snapshot of data at a certain time;
- Transactional replication – delivers transaction to subscriber;
- Merge replication – allows changing data at the publisher and at the subscriber, too.

The replication represents the process of copying and maintaining database objects in the database of distributed database systems.

4 Update of database application using replication

Let's have two database servers SQL_1 and SQL_2 with the databases data_1 on SQL_1 and data_2 on SQL_2.

We can prepare the necessary parts of database application, e. g. tables, views, stored procedures, user – defined functions on database server SQL_1 in the database data_1:

```
CREATE TABLE tab1(id INTEGER PRIMARY KEY, column1 …,
column2 …, …)

CREATE TABLE …

CREATE VIEW v_tab1 as SELECT … FROM tab1

CREATE VIEW …

CREATE FUNCTION func1(@param1 INTEGER, @param2 …, …)
RETURNS …
AS
BEGIN
DECLARE @prem1 INTEGER
DECLARE @prem2 …
…
SELECT @prem1 = (SELECT … FROM tab1 WHERE id = @param1)
…
RETURN @prem1
END

CREATE FUNCTION …

CREATE PROCEDURE proc1 @param1 INTEGER AS
…
… WHERE id = @param1
…
GO

CREATE PROCEDURE …
```

We can assume that the distribution database has been created.

The snapshot replication must be created **before inserting of data into tables or views**:

- include tables, views, stored procedures, user – defined functions into publication;
- setup of subscriber (subscribers) – database data_2 on database server SQL_2;
- start replication.

We can apply the following commands for updating of database application:

```
ALTER FUNCTION func1 …

ALTER PROCEDURE proc1 …
```

The replication can be created after the update of database application:

- include **changed stored procedures, user – defined functions only** into new publication;
- setup of subscriber (subscribers) – database data_2 on database server SQL_2;
- start replication.

5 Conclusions

This paper reviews possibilities of replication for updating of some types of database applications.

These are cases when the application logic is realized by means of stored procedures and user – defined functions and each of the local databases is controlled by the same DataBase Management System (the homogenous distributed database system).

References

1. KNIGHT, B.: Microsoft SQL Server 2000 Pokročilé techniky, Computer Press Brno, 2004, ISBN 80-251-0111-8, 477 s.
2. MATIAŠKO, K. – VAJSOVÁ, M. – ZÁBOVSKÝ, M. – CHOCHLÍK, M.: Databázové systémy. Databázové technológie a aplikácie. EDIS – vydavateľstvo ŽU, Žilina 2008, ISBN 978-80-8070-821-4, 518 s.
3. SEMANČÍK, Ľ.: Databázové systémy. Vojenská akadémia Liptovský Mikuláš, 2004, ISBN 80–8040–230–2, 115 s.
4. SEMANČÍK, Ľ. – DEDERA, Ľ.: Distribuované spracovanie údajov v databázových aplikáciách. Učebný text. Akadémia ozbrojených síl generála Milana Rastislava Štefánika Liptovský Mikuláš, 2010, ISBN 978-80-8040-411-6, 67 s.
5. TASCHNER, K.: *Nebojte se databází.* In: Computer World, č.44/96, s.17-30.