

THE SOLUTION OF REDUNDANCY PROBLEM IN DISTRIBUTED CONTROL SYSTEMS WITH HOT BACKUP OF DATA ACCESS SERVER

Darko ČAPKO, Aleksandar ERDELJAN,
Dragan S. POPOVIĆ, Imre LENDAK

FACULTY OF TECHNICAL SCIENCES, UNIVERSITY OF NOVI SAD

ABSTRACT

Distributed Control Systems often includes Data Access Server that mediates in communication between clients and devices. In order to improve reliability, the system is extended with one more (backup) server machine, which enables redundancy of the DA Server. Thus, there are two DA servers in the system, primary and backup. Clients application will be continue activity in communication with the new primary DA Server. The point is that clients don't communicate with DA Server directly than through an additive component – DARB. Role of the DARB is accompaniment DA Server status in the system and mediates in communication between clients and DA servers. This paper describes DARB component.

KEY WORDS:

Distributed Control Systems, Data Access servers, COM, CORBA

1. INTRODUCTION

Evolution of Distributed Control Systems and SCADA systems is qualifying of requirement that Data Access (DA) Servers are integrated in theirs. Nevertheless, data sources for DA Servers are PLC/RTUs or the other SCADA systems. Introduction DA Server in a system enables uniform accessing to devices, and according to this, numerous standards for DA servers are established. The most popular are OPC and DAIS. OPC standard enables access to the DA Server through COM (*Component Object Model*) interfaces, and OPC clients and servers are running in Windows environment. DAIS, on the other hand, enacts CORBA (*Common Object Request Broker Architecture*) interfaces, wherewith clients and servers can running either or in Windows or in UNIX environment.

In order to to improve reliability, the hardware and the software redundancy are intreducted in the system. The hardware redundancy implies duplicating computer machines and network (communication) lines. The software redundancy implies that there are two SCADAs and two DA Server in the system: primary and backup.

Klients are accessing to DA Server and upon these communications get information about the system. DA Servers are created according to

existing standards, but not totally. The point is that DA Servers absolutely make failover within viewed the system. Clients have to communicate with both servers and can check statuses of servers through it.

This paper describes the component that is on the client side and enables connecting with DA servers to clients. This component is named DARB (*Data Access Redundancy Broker*). Its assignment is enabling to client communication with DA Servers. If servers change their status (failover is happend) DARB switches to new primary server and continue communication with them. Client implementation is simplified herewith, because necessary activities are relinquished (if failover is happend) from client to DARB. Additionaly advantage in using of DARB is connecting to client that is running on Windows operating system (COM client) to DA Servers on UNIX (CORBA server).

2. THE DATA ACCESS REDUNDANCY BROKER COMPONENT

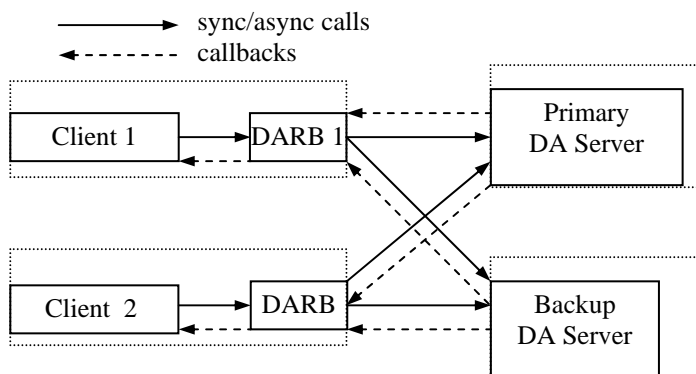
Client application exchange information with the primary DA Server while it is active. When the client finds that the primary DA Server isn't active (for example, communication breakdown, primary machine is offline, etc.), it should automatically switch to the backup server, treating it as primary. After that, when the primary server is coming online, it is declared as backup, and the clients are informed about it. Clients don't communicate with DA Servers directly, but via the DARB component. Each client creates its own DARB component on the client machine.

The DARB component has two roles in a DCS systems. DARB is, for first, server for client application which should access data in the DA Server. Second, DARB is a client for DA Server and hides existing DA Server from the client application. The basic task of the DARB component is to provide data exchange between the client and the primary DA Server. Accordingly, existing of a backup server is hidden for a client. Additionally, activity of DARB could be described through following functions:

- it monitors the state of primary DA Server and if it is RUNNING, forwards requests from the client to the server,
- if primary server isn't active (RUNNING), DARB switches activity to the backup server and observes him as primary server;
- acquire and stores interfaces of primary and backup DA Server;
- provides for backup server restoring of client requests to primary server;
- if client should need to know the state of the servers in the system, DARB gives access to this information;

Location of the DARB component in the system is presented on picture 1.

As illustrated in the picture above, commucation between components in a system is synchronous and asynchronous. Clients send requests to DARB for async callbacks and it forwards them to the primary DA Server. When some measurement is changed, DA Server advises DARB and it advises client about changes.

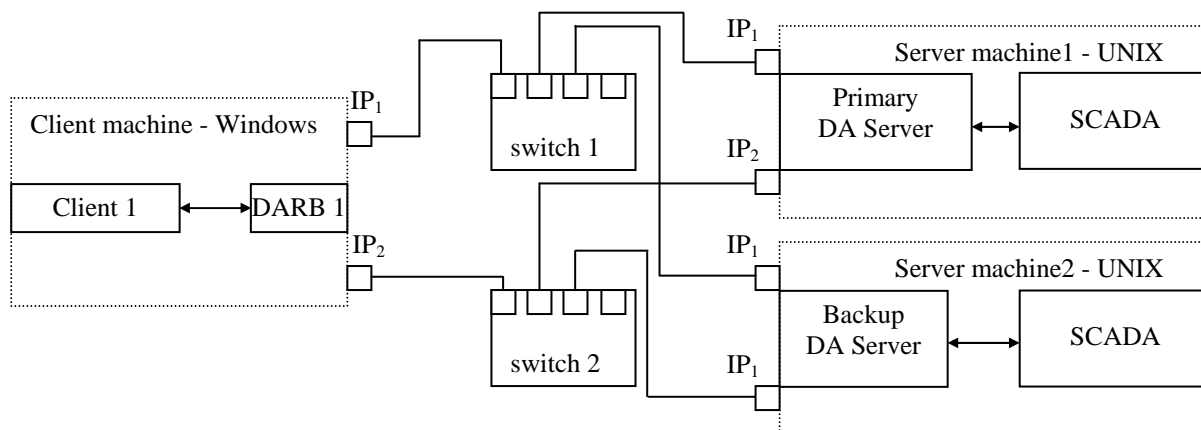


Picture 1. Location the DARB component in system

Configuration of the system is following:

- there are two networks in the system;
- all machines (clients and servers) are on two networks – have two IP addresses;
- communication between components in the system is enabled either via first or via second network.

View of the considered system is presented on picture 2.



Picture 2. View of the considered system

Therefore, for example, when network 1 is down on Server machine 1, communication between client (DARB) and primary DA Server is running via network 2. If a client can't communicate with the primary DA Server it changes primary server (treats server 2 as primary).

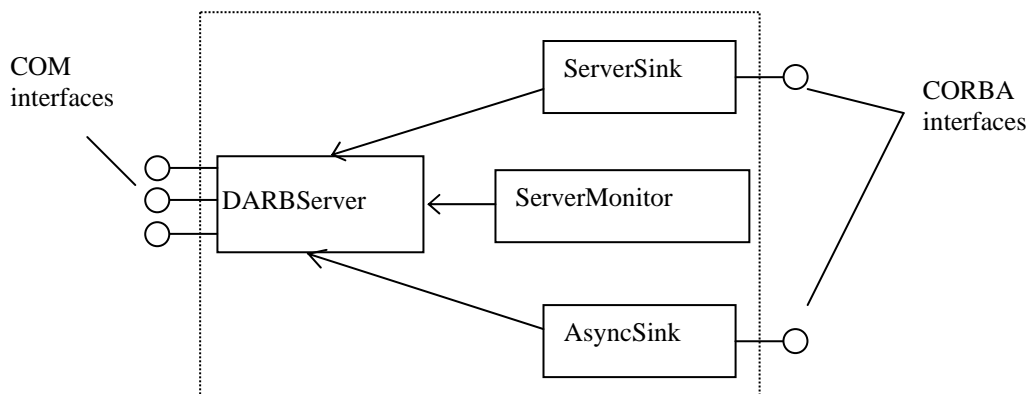
DA Server set own state according to SCADA state, thus when a SCADA changes state from primary to backup then appropriate DA Server change state.

3. ARCHITECTURE OF THE DARB COMPONENT

DARB component consist of a few objects: DARBServer, ServerSink, AsyncSink and ServerMonitor. Architecture of the DARB is presented on picture 3.

Major object in the DARB is the DARBServer which receives requests of the client, arranges datas and sends request to primary DA Server. After that, primary DA Server processes received requests and sends results of operation to DARB. A received results DARB sends to client.

DARBServer has COM interfaces on the client side, and accesses to DA Servers via CORBA interfaces.



Picture 3. Architecture of the DARB

Client via DARBServer creates objects ServerSink and AsyncSink that serve to async operation. When is some data changed on a DA Server, it send callback about this to AsyncSink object and DARBServer send this information to client. DA Server notifies all clients when it shuts down through sending callback to ServerSink object, and if backup server won't become, each primary DARBServer send shut down request to client.

ServerMonitor is object in a DARB that check a state of primary and backup servers in a system. If the state of primary server isn't RUNNING (primary) ServerMonitor get the state of backup server. If backup server is RUNNING it is declared as primary and on the other hand, DARBServer send shut down request to client.

4. IMPLEMENTATION

DARB is component that forwards information between client and primary DA Server and aims its flow.

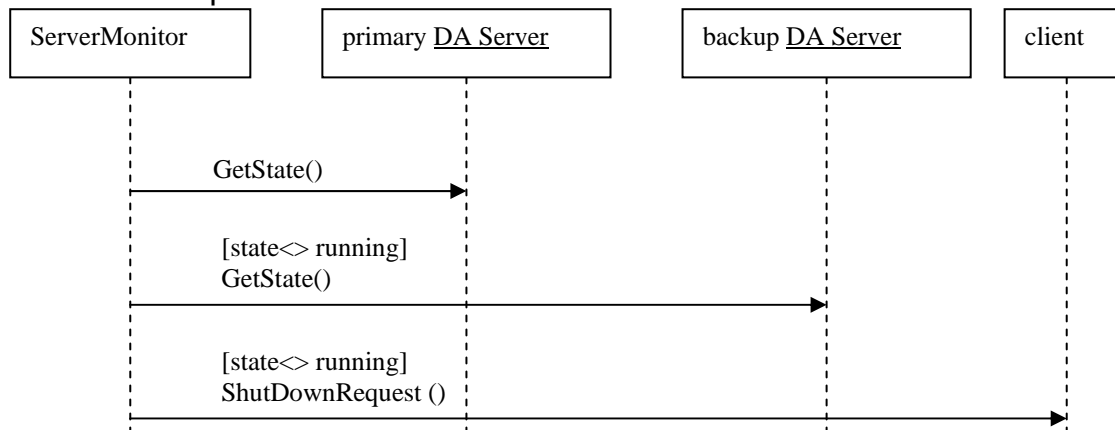
It accesses to DA Servers via omniORB which is an Object Request Broker (ORB) that implements CORBA 2.6 specification. DARB is running on Windows operating system and by omniORB connects to DA Servers that are running on UNIX operating system. On the other hand, client is running on the same machine as DARB and it is connected to DARB via COM interfaces. DARB is dll component that is running when a client is active.

For example, picture 4. shows iteration diagram working of the ServerMonitor component.

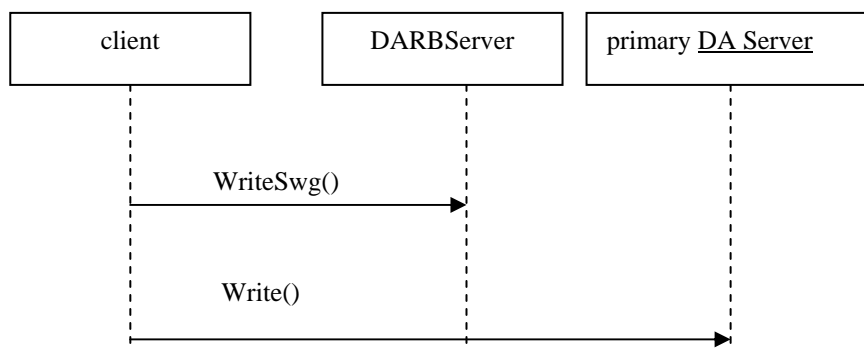
ServerMonitor calls GetState method on primary DA Server and if state of server isn't RUNNING, ServerMonitor checks state of backup server. Provided that, connection with primary server machine through IP1 is broken, ServerMonitor repeats call through second network (IP2). If ServerMonitor can't get state of primary server through IP2, it calls this method on backup DA Server (through both network, if necessary) and if its state isn't RUNNING, ServerMonitor send to client ShutdownRequest().

Similarly, when client calls, for example, WriteSwg method (change value of some switch), DARB is that call proceeds to DA Server. When DA

Server arranges request of DARB, preceeds result of the operation to DARB that send result to client. Iteration diagram write operation is presented on picture 5.

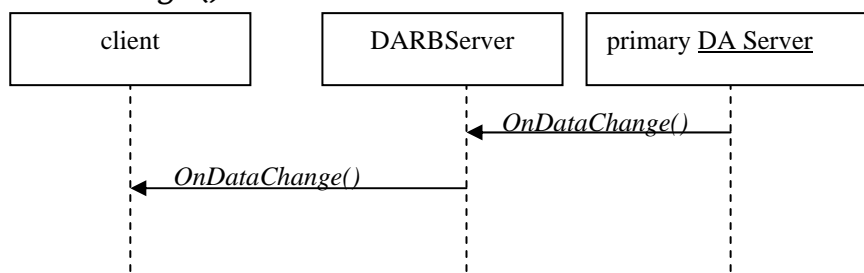


Picture 4. Iteration diagram working of the ServerMonitor component



Picture 5. Iteration diagram of the Write method

Client will be informed if value of any item is changed by the callback mechanism. DA Server gets information about of changing some items and send this information to AsyncSink object of the DARB that is send *OnDataChange()* to client. Picture 6. illustrates callback mechanism.



Picture 6. Iteration diagram of the OnDataChange method

Client machine, also have two IP addresses and DA Server sends callbacks via available network. During a runing, DARB send information to client about of state of servers in the system.

5. CONCLUSION

Prividng requirement to increase reliability of a Distributed Control Systems, sense parts of system are agitated and analysed possible solutions. DCS often have two SCADAs: primary and backup, which

permits access to clients in depends of his role in system. In SCADA systems, an DA Server often represents a “bottleneck” or single point of failure. Therefore, there is a need for the backup of DA Server, that will be activated (put in running mode) if some failure happens in the primary server, apropos each DA Server follow behaviour appropriate SCADA and determinates own status in system. Backup DA Server is identical as primary DA Server.

On client side, DARB object is introduced to monitor primary and backup server activity. It implements COM interfaces and acts as a mediator between client and primary DA server. Task of this component is to check status of DA Servers and receiving requests of the client. Additionally, DARB is a client to DA Server because it is send appropriate requests (received from client) to DA server. When primary DA Server accomplishes established tasks, results are sends to the client.

REFERENCES

- [1.] Dale E. Rogerson, *Inside COM*, Microsoft Press, Redmond, 1997.
- [2.] Darko Capko, Aleksandar Erdeljan, *Hot backup of OPC Data Access Server*, XLVII ETRAN, 2003.
- [3.] *OPC Redundancy Broker, Users' Manual, Version 1.1*, Matricon, February 2001.
- [4.] “Common Object Request Broker Architecture (CORBA) 2.6”, OMG specification, <http://www.omg.org/omniORB> (version 4.0.5) documentation,
- [5.] <http://omniorb.sourceforge.net>