ASYNCHRONOUS DATA CHANGE NOTIFICATION BETWEEN DATABASE SERVER AND ACCELERATOR CONTROL SYSTEMS*

Wenge Fu, Seth Nemesure, John Morris, Brookhaven National Laboratory, Upton, NY 11973, USA

Abstract

Database data change notification (DCN) is a commonly used feature. Not all database management systems (DBMS) provide an explicit DCN mechanism. Even for those DBMS's which support DCN (such as Oracle and MS SQL server), some server side and/or client side programming may be required to make the DCN system work. This makes the setup of DCN between database server and interested clients tedious and time consuming. In accelerator control systems, there are many well established software client/server architectures (such as CDEV, EPICS, and ADO) that can be used to implement data reflection servers that transfer data asynchronously to any client using the standard SET/GET API. This paper describes a method for using such a data reflection server to set up asynchronous DCN (ADCN) between a DBMS and clients. This method works well for all DBMS systems which provide database trigger functionality.

INTRODUCTION

In normal database server and client operation, the server side stores the data and makes it accessible to clients. Clients dominate the data changes by submitting requests to the server. There are many instances where a client may want notification when the data has changed on the server side by another client. There are two known ways to accomplish this task. One way is to use a database server driven data change notification (DCN) technique while an alternative approach is to have the client synchronously poll for data. The downside to client data polling is an increased (and potentially unnecessary) number of network transactions in addition to an increase in database server side work load. Additionally, depending on the data polling frequency, there may be a delay between the time that the client becomes aware of an update to data and the time the update actually took place. By contrast, database server side DCN provides the client with asynchronous (immediate) updates.

The distributed DCN message structure can be in the form of the actual changed data or can be a form of meta data (e.g. the column in the database table that has changed including whether or not the change was due to a database insert or update). In recent years, many commonly used DBMS (Database Management Systems) such as the Oracle database server[1], Microsoft SQL server[2], IBM DB2 server[3], and Sybase ASE[4] server, provide DCN support using different approaches. There are advantages and disadvantages to each DBMS system's DCN feature. Typically, in order to take advantage of a DBMS provided DCN feature the user may need to:

- Set up a data change notification handler on the server side for target data tables or target data columns, and publish data changes (or event or meta data of the data changes without actually sending the changed data) through the handler;
- Set up a data change notification receive handler on the client side and register (or subscribe) it with the listening DBMS DCN objects that will receive the data change notification (or events and meta data of the data changes) .
- Keep an open (TCP or UDP) network connection between client and database server.

Setup of DCN registration, data change publishing, and receiving DCN may differ between various DBMS to DBMS systems. At a minimum the basic steps listed above must be taken into account. Some database server side SQL programming and client side application programming may also be required to make the system work. This process can be tedious and may require additional effort to support different clients.

In typical accelerator control systems like the RHIC-AGS system running at Brookhaven National Laboratory, there exist common techniques for data communication between hardware and software. Some of these include software interfaces such as ADO[5] (Accelerator Device Object), CDEV [6] services, and EPICS[7] (Experimental Physics and Industrial Control System). These interfaces can be used to setup reflection servers to transfer asynchronous DCN messages from a database server to client without a direct connection between client and database server. This greatly simplifies the setup of the DCN process.

By taking advantage of existing accelerator control system software using the CDEV service, an example of a simple and robust method for setting up a database DCN system is presented. This method works well with any DBMS system which supports database triggers.

SETUP OF DATABASE ADCN

The ADCN system consists of three parts as shown in the Figure 1.

* Work supported by Brookhaven Science Associates, LLC under Contract No. DE-AC02-98CH10886 with the U.S. Department of Energy.

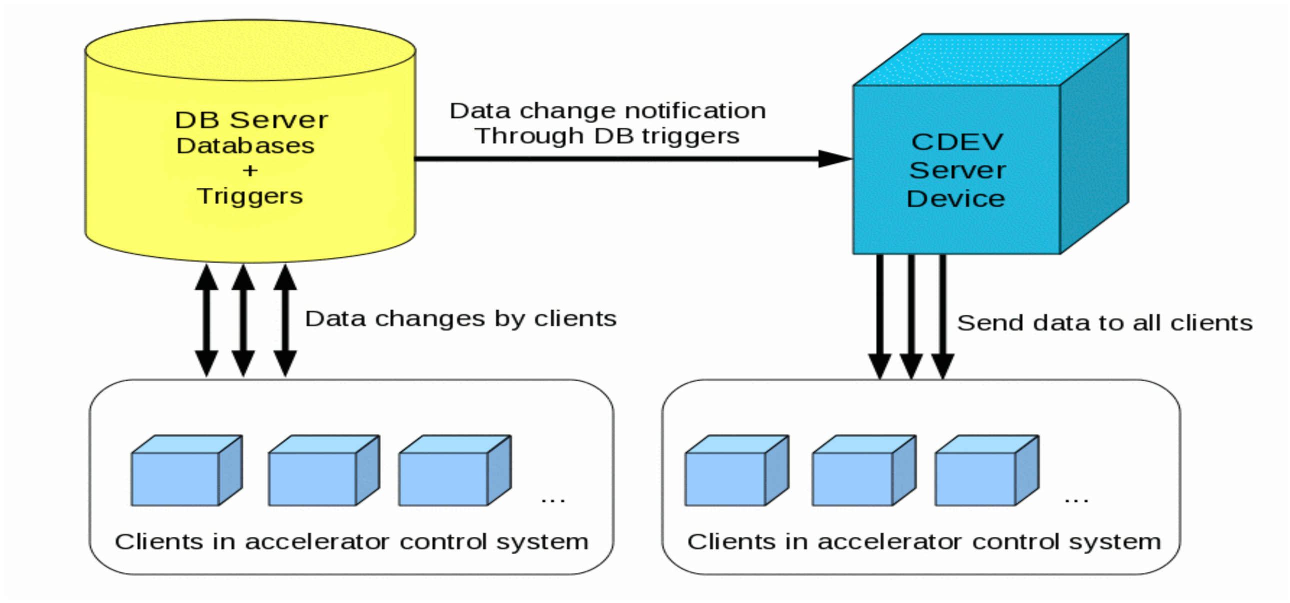

Figure 1: Diagram of ADCN system setup.

1. A database which stores the controls data. The database can exist on any DBMS system which supports the database trigger feature and able to execute system commands from the database trigger. Some DBMS examples are Oracle, MS SQL Server, Sybase ASE, and MySQL. In a target database, a specially designed trigger can be set up to fire whenever the target control data changes. The data changes can be in the form of newly inserted data, updated data, or removal of existing data. Depending on the capability of the DBMS system, the trigger can fire before or after the data changes in the database. The database trigger can also be designed to determine what kind of DCN message to send. The developer set the system up so that it sends events of data changes, meta data of the changed data, or the changed data itself. In this case the trigger will send the DCN message immediately along with the data change that occurred in the database.
2. A second requirement is for the existence of a generic server that listens for the trigger and transmits the DCN message from the database to interested clients. The generic server here can be any well established accelerator control system server such as an ADO or CDEV Server. In the example above CDEV server devices are used.
3. The third requirement is for a client to set up the communication exchange with the server. This allows the client get notified when the target data changes initiated by other clients. The data change process between client and server is independent from these clients of being notified. At the time of a DCN, the data within the database table is updated which initiates the asynchronous transaction to interested clients. In addition to being a place for control data storage, the database becomes integrated into the accelerator control system.

The key component of this system is the design of the database trigger. This is the only part that requires knowledge of SQL programming. The trigger requires the user to determine the type of data to send to clients and under what conditions. Depending on the DBMS system used, the execution of a trigger may or may not affect the database operation in the case where the trigger execution fails. It is good practice for the developer to make sure the trigger has the flexibility to support various failure modes.

In the ADCN system shown in Figure 1, the set up of the CDEV server device is straight forward due to the developers familiarity with well established CDEV (ADO) interface. The CDEV server device serves as a reflective device (the server only needs to forward data from the database server to interested clients). The server supports the same API (i.e. SET/GET methods) used by other devices in the accelerator control system. In addition to specialized applications that use the ADCN system in the Collider-Accelerator Controls System at Brookhaven National Laboratory, generic wide use applications such as GPM (Generic Parameter Monitor), pet (Parameter Edit Tool), and the logging system can also take advantage of ADCN.

ADCN SYSTEMS USED IN RHIC

In the RHIC-AGS controls system, several ADCN systems have been set up by using the methods described in this paper and have been used operationally over the past several years. The DBMS systems involved in these ADCN systems are MS SQL Server 2005 running on a Window PC and Sybase ASE 15 running on Solaris 10.

The same ADCN set up strategy can be applied to any DBMS system which supports the trigger feature. Below is a brief description of one of the ADCN system--- the RHIC access controls system for live IRIS data communication.

In the RHIC access controls system, IRIS recognition data are generated from the IRIS access control devices and stored in a MS SQL Server 2005 database on a security sub-network on a Windows PC. The MCR (main controls room) of RHIC requires asynchronous notification whenever IRIS data in the database is updated. To achieve this, a database trigger was created in the MSSQL database. The trigger initiates the transfer of the changed data to the pre-defined CDEV server running on a Linux host. The MCR application monitors and logs the data from the CDEV device and uses the live IRIS data change for security access in the CAD complex. Figure 2 shows how the system works.

In this example, the ADCN system seamlessly integrates the IRIS access control system (manufactured by LG) and MS SQL Server (2005) database with the in-house designed access control system.

SUMMARY

Asynchronous data change notification (ADCN) between database server and clients can be realized by combining the use of a database trigger mechanism, which is supported by major DBMS systems, with server processes that use client/server software architectures that are familiar in the accelerator controls community (such as EPICS, CDEV or ADO). This approach makes the ADCN system easy to set up and integrate into an accelerator controls system. Several ADCN systems have been set up and used in the RHIC-AGS controls system.

REFERENCES

[1] Oracle@ Database New Feature Guide: 11g Release 1(11.1), B28279-03, October 2008.

[2] Microsoft SQL Server: http://msdn.microsoft.com/en-us/library/ms130214%28v=sql.100%29.aspx

[3] BM DB2: http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp

[4] Sybase: http://infocenter.sybase.com/

[5] D.S. Barton et al. RHIC Control System, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment,* Volume 499, Issues 2-3, 2003, Pages 356-371.

[6] CDEV:http://www-csr.bessy.de/control/Soft/cdev/cdevIntroduction.html

[7] EPICS: http://www.aps.anl.gov/epics/

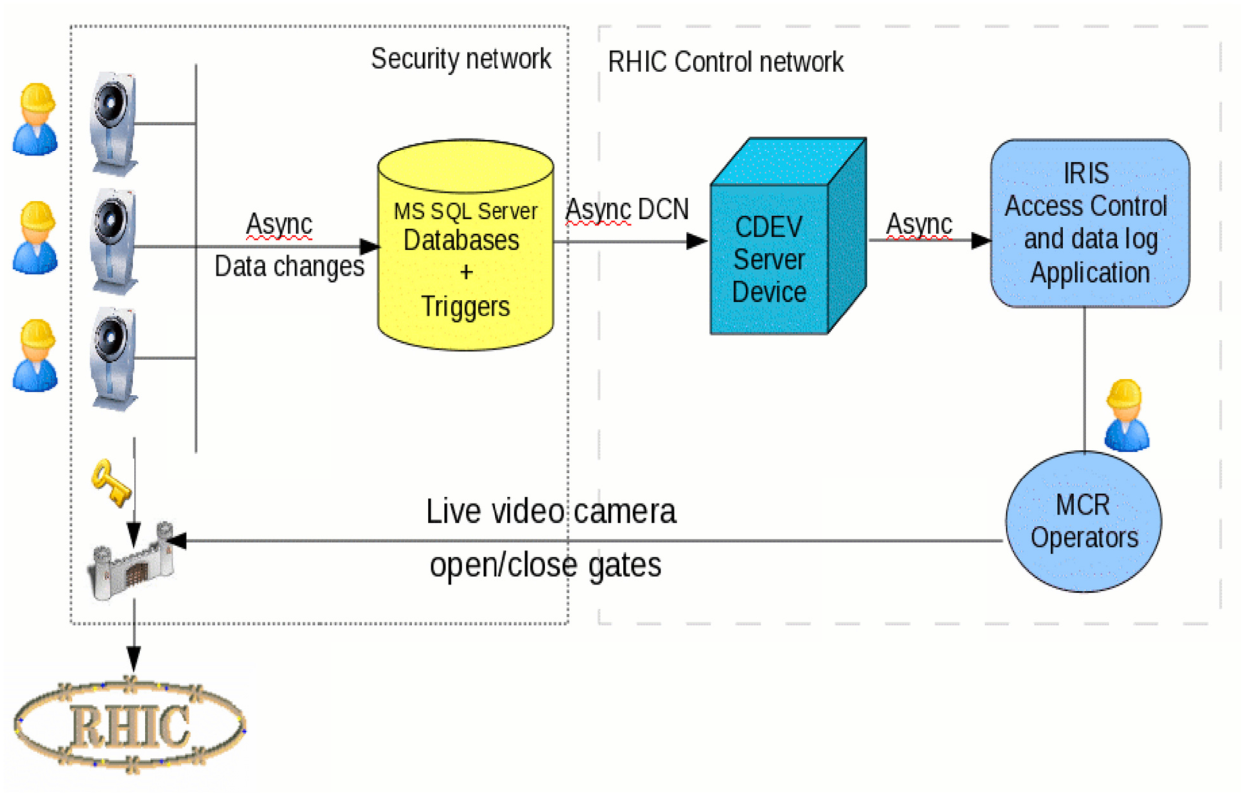

Figure 2: Diagram of RHIC IRIS Access Control System with ADCN.