Description

PoC at Microsoft

2023-04-04

1 Replication Scenario 2
2 Environment 2
3 General 3
4 tcVISION Agent Installation 3
4.1 tcVISION Agent in z/OS 3
4.1.1 tcVISION Agent Installation Requirements 3
4.2 Using Db2 UDT “agentless” 3
4.3 tcVISION Server Agent 5
4.3.1 Kafka connection 6
5 TCP/IP 6
6 tcVISION Design Studio 6
7 tcVISION Repository 7
8 Staff 7

1 Replication Scenario

The Trial should implement the general functionality of tcVISION. The criteria for the installation are defined as setup of a successful, repeatable, automated replication process with documented results from z/OS Db2 to Kafka[1]. The trail will be held for 30 days.

A few Db2 tables will be used for replication. These tables will be transferred to the target database using the tcVISION bulk functionality and then replicated to the Kafka target using the tcVISION real time function. Each message will have an “action” flag and a unique key to identify modified or removed records.
The throughput and the reliability will be considered in particular. It is assumed that the test environment has the technical capacity to achieve the latency specifications.

2 Environment

tcVISION Mainframe Agent[2]:	z/OS V2.x.0
tcVISION Source Databases:	Db2 V12 or V13 on z/OS
tcVISION Server Agent:	Windows or Linux
tcVISION Target Database:	Kafka on premise or Azure Event Hub
tcVISION Design Studio:	Any JAVA capable platform

1 Or Azure Event Hub

2 If needed, the “agentless” solution works without any zOS region.

3 General

tcVISION can synchronize Db2 changes made online or via batch in real time via the Db2 log. To keep the impact on the mainframe as low as possible, tcVISION sends the changes directly to the replication server. The whole processing for initial load and real-time capturing can be automated.

4 tcVISION Agent Installation

4.1 tcVISION Agent in z/OS

It is possible to use the tcVISION function Db2 UDT agentless for pure Db2 processing without the host agent.

The tcVISION Agent

- collect the changed data from Db2 via the Db2 Active Logs
- control the running transfer scripts

A detailed description of the mainframe installation under z/OS can be found in chapter 2 of the manual "tcVISION Host Installation and Administration". The Db2 access will be described in chapter 4.1. Further information can be found in the manual "tcVISION tcSCRIPT" starting in chapter 8.3.

The tcVISION Mainframe Offload Data feature can also be used for the Bulk method, where the bulk is done via Db2 Imagecopy on the tcVISION Replication Server. This functionality does not consume any resources on the mainframe, because the complete processing is done on the platform of the tcVISION replication server.

4.1.1 tcVISION Agent Installation Requirements

The tcVISION Agent z/OS needs its own region.

The tcVISION modules, job samples, and macros require the tcVISION installation library. All required jobs can be found on the installations library.

The tcVISION Agent requires access to all needed resources on the mainframe.

4.2 Using Db2 UDT “agentless”

The capturing of changes from the active log of Db2 on z/OS can be done using a User Defined Table. With this type of processing no active tcVISION component (Host Agent) is required to run in z/OS. This

processing is possible in Db2 version 10 with fix PM90568 and in Db2 version 11 or higher without any restrictions.

The tcVISION UDT is defined with the following CREATE statement:

```
CREATE FUNCTION TCVUDT_V7 (VARBINARY(300),VARBINARY(32000))
RETURNS TABLE (OUTVALUE BLOB(200K))
PARAMETER CCSID EBCDIC
EXTERNAL NAME TCSD2UDT
LANGUAGE ASSEMBLE
PROGRAM TYPE SUB
STAY RESIDENT YES
CONTINUE AFTER FAILURE
WLM ENVIRONMENT wlmenv
RUN OPTIONS
'H(,,ANY),STAC(,,ANY,),STO(,,,4K),BE(4K,,),LIBS(4K,,),ALL31(ON)'
PARAMETER STYLE SQL
NO SQL
NO EXTERNAL ACTION
FENCED
SCRATCHPAD
FINAL CALL
DISALLOW PARALLEL;
```

wlmenv is one of the defined Application Environments in the z/OS WorkLoadManager (WLM).
The tcVISION loadlib containing the module TCSD2UDT needs to be added to the JCL of the Application Environment defined procedure in the STEPLIB concatenation. All loadlibs of the steplib concatenation need APF authorization because of the used IFI calls.

The user Id of the calling script requires the following Db2 authorizations:

```
GRANT EXECUTE ON FUNCTION TCVUDT_V7 TO userid
GRANT MONITOR2 TO userid
```

The definition of the calling scripts on the workstation is now Input_Source_Type = 'IFI_306'.
Input_Source_Name contains a connection string for DRDA or Db2 Client to the z/OS Db2.
For both connection types, a specific database name (e.g. Db2 subsystem name instead of the Db2 location name) that should be used for further processing (e.g. database in repository, etc.) can be specified with the optional parameter SUBSYS=.

If the creator of the tcVISION UDT does not correspond to the Userid of the calling script, the full name of the tcVISION UDT together with the parameter UDT_Name has to be specified, for example:

```
UDT_Name = 'CREATOR.TCVUDT_V7'
```

If an update of tcVISION UDT is necessary, the module on the LOADLIB has to be changed and then reloaded:

```
/VARY WLM,APPLENV=environment-name,REFRESH
```

The parameter IFI306_COMMIT_INTERVAL is used to limit the reading process of the UDT data in time. After this interval in seconds, the Stored Procedure sends the data that was read up until then to the UDT client. Values from 1 to 60 seconds are valid. For the UDT Stored Procedure, a value from 5 to 10 seconds is recommended:

```
IFI306_COMMIT_INTERVAL = 5
```

With the parameter DB2_UDT_MAX_BLOCKS, the reading of UDT data is also limited in terms of volume . After reaching the number of blocks, the Stored Procedure sends the data read up to that point to the UDT client. Values from 0 to 5000 are permitted, whereas the value 0 does not lead to a limitation:

```
DB2_UDT_MAX_BLOCKS = 500
```

4.3 tcVISION Server Agent

The tcVISION on the server will receive the changed data from z/OS and apply the data to the target system(s). A standard installation procedure will install the components to the server. The installation requires 20 MB disk space. If data is stored and buffered on the server, more disk space might be needed.

For Linux and UNIX the unixODBC package must be installed. Optional components may be required if additional functionality is necessary.
For Windows environments the “librdkafka” together with OpenSSL must be installed.

The tcVISION Agent can be installed on premise or in a cloud instance. The communication from tcVISION to the databases is established as follows:

- Db2 IBM Client or built-in DRDA driver to Db2
- “librdkafka” to Kafka or Azure Event Hub

For the PoC we recommend a Windows or Linux installation. tcVISION runs on the following operating systems:

- MS Windows 64-bit
- Linux 64-bit
- Linux on IBM Z
- IBM AIX 6.x and later; IBM PowerPCs
- sun Solaris 9 and later or HP UNIX

The tcVISION Repository can either reside on the server with the target database or on any other server of choice outside the host.
The server manager must not run as root.

4.3.1 Kafka connection

Setting up a connection to the target "Kafka" or Azure Event Hub is described in the chapter "5.3.4 Connection to Azure" and "5.3.5 Connection to BigData".
The library "librdkafka" and OpenSSL is mandatory.

5 TCP/IP

The tcVISION components must be able to connect to each other. Connections must be possible:

- from z/OS Agent and scripts to the server
- from server Agent and scripts to the mainframe
- from tcVISION Design Studio to both the server on which tcVISION is running and to the mainframe

All tcVISION components must be able to connect to each other in both directions. To ensure the connectivity, at least ten ports must be available. The number of ports to reserve depends on the number of parallel tasks desired to run in tcVISION. Please adapt the firewall openings.

6 tcVISION Design Studio

The tcVISION Design Studio will be used to monitor, administer, and control the different tcVISION agent and replication processes. It runs on any Java capable platform.

The tcVISION Design Studio is to be installed can run on any JAVA compatible platform. For the installation a directory with free space of 40 MB is required. All writing operations will be placed into the users directory.

The tcVISION Design Studio is not necessary for the transfer itself. It will be used for defining and visually monitoring the transfer process only.

7 tcVISION Repository

The tcVISION Repository is the source for meta information about input and output objects. All replications performed by tcVISION must be defined to the Repository.

The tcVISION Repository consists of tables in a database. Supported databases are:

- MS SQL Server
- PostgreSQL
- MySQL / MariaDB
- Db2
- Oracle
- EXASOL

All Agents in a tcVISION network access the same database. If it is not possible for an Agent to directly access the database, the Repository access can be redirected to another Agent in the network with direct access. This also applies to scripts.

Currently, support is provided for the mainframe, MS-Windows, UNIX, and Linux platforms on which the tcVISION Agent can run. When choosing the database that should be used for the tcVISION Repository, the main focus should be on platforms on which the DML statements reside.

8 Staff

For the installation we recommend that the following people are available:

- z/OS system administrator
- Database administrator
- Network administrator[3]
- Windows/Linux administrator
- Staff who is familiar with the data structure and content of the databases

3 Especially when running with Azure Event Hub