Maximize Your Return on Big Data

May 15-18, 2012
Scalable and Reliable Real-Time Data Integration Using Informatica CDC

Allen Bailey & Victor Kennedy
ING GROEP NV
Presenters

Allen Bailey, Mgr ETL & Data Warehousing

- Working with Informatica PowerCenter since v3
- OLAP & Dimensional Data Modeling
- Data Movement (Extract, Transform, Load) Architectures

Victor Kennedy, Lead ETL Engineer

- Working with Informatica PowerCenter since v4
- Subject Matter Expert (SME) for CDC and Traditional ETL
- Informatica Certified Consultant v8
Session Objectives

Informatica CDC Architectural Detail

• What Change Data Capture (CDC) is
• What CDC does
• How it does it

Alternatives to CDC

• RDBMS Replication
• Database triggers and standard ETL
• Traditional batch ETL
Session Objectives

Key Decision Points

• RDBMS capacity for transaction processing
• Latency requirements
• Heterogeneous Targets (different RDBMS vendors)
• Data structure similarity from source to target

Implementation and Production Support

• Installation of add-on components
• Efficiency of CDC Read vs. Load Balancing in PowerCenter
• Token structure & management
• Restart & Recovery considerations
Application Architecture (Prior to CDC)

- Policy Admin at 100% CPU (8 Dual Core)
- Database outages due to heavy usage
- Adversely impacting 500+ users
Application Architecture (After CDC)

• Split applications across two databases on two separate Servers.
Alternatives to Informatica CDC

RDBMS Replication
- RDBMS vendor supplied database replication utility
- Third Party supplied database replication utility
- Limited scope control but fast and robust
- Expects source and target schemas to be identical

Database triggers and standard ETL
- Triggers based on the Insert/Update or Delete transactions on a table-by-table basis.
- Copies change data to another table for ETL to source from
- Can embed business logic to determine when to copy or not
Alternatives to Informatica CDC

Traditional batch ETL

• Extraction based on Last_Updated_TS as compared to the batch control timestamp from the last cycle.
• Error handling and recovery is based on timestamp.
• Using standard database connectors for “vanilla” PowerCenter mappings.
• Periodic; i.e. nightly, weekly, monthly batches
• Total programmatic control of source to target field mapping
• Extraction and Transformation/Load can be decoupled.
• Latency is dependent on the length of the period between cycles.
Key Decision Points: CDC vs. Alternative 1

RDBMS capacity for transaction processing
• Can the database server handle additional SQL?

Latency Requirements
• Is batch possible; i.e. will a cycle complete before the Service Level Agreement (SLA) expires?

Data Structure similarity from source to target
• If the source and target schemas are not identical, you may have to use CDC. Varies by vendor.
Key Decision Points: CDC vs. Alternative 2

Heterogeneous RDBMS Targets

- If the source RDBMS and target RDBMS are different products or vendors, you may have to use CDC.
Informatica CDC architectural detail (1)
Informatica CDC Architectural Detail (2)

What Informatica CDC Power Exchange is

- A “reader” for the RDBMS log sub-system (via API)
- It is not a stand-alone product. It extends PowerCenter.
- It has its own data mapping development environment

What CDC Does

- It calls the RDBMS log read API using “Tokens” that control the start points within the appropriate log file at a table level.
- It provides the resulting stream of data to a PowerCenter mapping.
- It maintains current token values in the Target database and the Informatica server.
How it does it

• A “listener” agent on the source RDBMS server connects to the RDBMS’s logging API; e.g. DB2 Log Read API.
• When PowerCenter issues a read request the listener passes the starting token value to the API.
• As the API returns transaction records to the listener, the listener streams these into the calling PowerCenter mapping.
• From the point at which the data enters the PowerCenter mapping, processing is no different than it would be for a flat file or relational source.
• When the API has reached the end of the log, it either switches to the next log or waits for arriving transactions.
PowerExchange CDC Real-Time Platform
PowerExchange CDC Extraction Modes

- **Batch mode (CAPX)**
  Bulk Data movement in periodic batches on a specific time window.

- **Continuous mode (Continuous CAPX)**
  The change data is condensed and stored in the PowerCenter and moved continuously to the Target in near real time.

- **Real-Time mode (CAPXRT)**
  Change data is captured and moved in near zero latency.

Why we chose CAPXRT? Find out from the POC results

- 500 concurrent human user simulation using Load Runner.
- High OLTP overhead simulation on Source Database.
- Throughput 102 transactions per second by PWX CDC Ver. 8.6.1.
- 100% of all the changes applied to Target within 8 seconds using CAPXRT.
- Changes applied within 30 seconds using Continuous CAPX.

Note: PWX Condense component has been deprecated in Ver. 8.6.1 and future PowerExchange versions will work on PWX Logger only.
Phases of Change Propagation in Real-Time

- Capture
- Collate
- Cleanse
- Extract
- Transform
- Load

- PWX Virtual Change Stream
  - Collector
  - UOW Cleanser
  - Listener

- PowerCenter RT
  - dbmover.cfg
  - Token Files
  - Xmaps

- TARGET DATABASE

- PWX Navigator
- PWX Client Tool

- Navigator is a PWX client tool required for creating Extraction objects and checking changed data.

DB2/UDB
- DB2 Logs
  - DB2 Log Read API
The PWX DBMOVER configuration file contains parameters that govern and control PWX processes.

This configuration file is required on Integration Service Machine where PWX is installed and also on each system where PWX components are installed.

Whenever the PWX starts up, it first reads the “dbmover.cfg” for configuration information.

Informatica provides a sample configuration file for each supported platform. Simply edit the file to configure, and customize the CDC operations.

Dbmover Syntax

```plaintext
CAPI_CONNECTION=(
    NAME=name,
    TYPE=(UDB,
        [CCATALOG=capture_catlg,]
        [DBCONN=database_name,]
        [USERID=user_id]
        [EPWD=encrypted_password,]
        [MEMCACHE=cache_size,]
        [PASSWORD=password,]
        [RSTRADV=seconds,]
        [UPDINT=seconds,]
        [UPDREC=num_records,] )
    LOGPATH=/op/pwx/logs
    CAPT_PATH=/op/pwx/reg
    CAPT_XTRA=/op/pwx/extmaps
)```

PowerExchange Configuration Files
PowerExchange CDC Listener

Listener is responsible for

- Accepting SQL requests from CDC sessions and Client tools.
- Capturing Changes from the Source using extraction Maps.
- Handling communications from PWX Components and Listeners on other Nodes.

PWX Utilities for Listeners

- `dtlinfo` – to verify installation
- `dtllst` – to activate a Listener
- `dtlrexe` – to ping remote Listener

Configuring Listener

```
LISTENER=(LST01,TCPIP,2481)
NODE=(local,TCPIP,127.0.0.1,2481)
NODE=(INF01PROD,TCPIP,10.160.150.14,2484)
NODE=(INF01PFIX,TCPIP,10.160.144.196,2454)
NODE=(INF01ACCP,TCPIP,10.170.129.84,2984)
NODE=(INF01QA,TCPIP,172.18.243.84,2984)
NODE=(INFA01DEV,TCPIP,172.18.226.43,2983)
NODE=(INFA01LAB,TCPIP,172.18.226.44,2983)
PWXSOMAXCONN=128
MAXTASKS=20
POLLTIME=1000
TIMEOUTS=(300,1200,-1)
```
Navigator: An easy-to-use Graphical Interface Tool (GUI)

Salient Features of PowerExchange Navigator

• PWX needs valid extraction objects such as Extraction Maps for relational sources and Data Maps for non-relational sources to pull the correct data.
• Navigator runs on Windows and facilitates point-and-click development methods to generate Extraction objects.
• Facilitates real-time visibility to captured changes.

Use PowerExchange Navigator to

• Create, view, edit and delete Registration Group, Capture Registration, Extraction Maps and Data Maps.
• Perform Data row test to extract and preview the changed data.
• Manage Schema and table changes and audit CDC loads.
• You can also use the Navigator to generate Restart tokens.
• Issue LISTTASK command to see the list of tasks or STOPTASK to stop the tasks started by listener.
Create Extraction objects in easy steps

- Use Navigator to create and define extraction objects as per the requirement.
- Specify Source name, User-id and Password to create Registration Group.
- Select Source tables and define table columns to add Capture Registrations.
- Navigator automatically generates Extraction Maps.

<table>
<thead>
<tr>
<th>Tables</th>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbo Categories</td>
<td>dbo CustomerID</td>
</tr>
<tr>
<td>dbo CustomerDemo</td>
<td>dbo CompanyName</td>
</tr>
<tr>
<td>dbo CustomerDemographics</td>
<td>dbo ContactName</td>
</tr>
<tr>
<td>dbo Customers</td>
<td>dbo ContactTitle</td>
</tr>
<tr>
<td>dbo Employees</td>
<td>dbo Address</td>
</tr>
<tr>
<td>dbo EmployeeTerritories</td>
<td>dbo City</td>
</tr>
<tr>
<td>dbo Order Details</td>
<td>dbo Region</td>
</tr>
<tr>
<td>dbo Orders</td>
<td>dbo PostalCode</td>
</tr>
<tr>
<td>dbo Products</td>
<td>dbo Country</td>
</tr>
<tr>
<td>dbo Region</td>
<td>dbo Phone</td>
</tr>
<tr>
<td>dbo Suppliers</td>
<td>dbo Fax</td>
</tr>
<tr>
<td>dbo Territories</td>
<td></td>
</tr>
</tbody>
</table>
How to perform a Data Row Test?

- Select and open the Extraction Map from the navigation panel.
- Select File > Database Row Test from the Top Menu.
- The Database Row Test dialog box appears.
- Fill out the connection details and click Go.
Sources and Targets for CDC ETL Mappings

How to create PWX Sources and Targets?

- To create Source definitions, in the Source Analyzer menu, select Sources > import from PowerExchange
- To import Target objects, in the Target Designer, click Targets > import from PowerExchange
Sample ETL Mapping for CDC

- Note down the multiple Source-to-Target pipelines in the Mapping.
- Group Sourcing is using multiple Sources in the same Mapping.
Pros and Cons of CDC Mappings

**PROS**

- Facilitates Group Sourcing, so that Change data for multiple sources can be read in a single pass.
- Same changed data can be consumed by multiple mappings.
- PWX Generated Columns such as CAPXACTION, CAPXUOW and Column Level indicators such as DTL_CI columns can be used to control the records and columns to be processed.
- PWX parameters can be used in the mapping transformations such as Source Qualifier, Expression, Update Strategy, Filter and Router.

**CONS**

- Cannot use transformation objects such as Aggregator, Joiner, Normalizer, Rank, Sorter and Constraint based loading.
- Additional steps involved in reading changed data from partitioned databases.
Configuring CDC Application Connections

Specify Values for

- Listener location
- UID & PWD
- Restart Token File Folder
- Recovery Cache Folder
- Idle time
- Maximum rows per Commit
- Real-Time Flush Latency

Application Name: Specify unique name. The default is the first 20 characters of the Workflow name.

Token File Name: Cannot be shared with other sessions. Token files are maintained under the Application name.
CDC Sessions and Workflows

Identify and configure the Key Parameters

- Change Commit type to Source based commit
- Disable Commit on End of File
- Decide on Unit Of Work (UOW) size and count per Commit
- Choose the Real-Time Flush Latency time
- Change Recovery Strategy to Resume from last Checkpoint
- Choose the value for Idle Time. Set the value to 0, to force the session to stop when it reaches end of log. The default value is (-1) which keeps the Session running continuously.
- Specify Unique names for Applications and Token Files.
PowerExchange CDC Token Files

How are Token Files used in Change Data Capture?

- PWX uses tokens to determine the exact point and position to restart.
- Tokens originate from the Source, and stay in the PowerCenter Cache until the changes are applied to Target.
- Session commits tokens into the Target along with the changed data.
- Both tokens and changed data are applied within the same commit scope.
- The Tokens are stored in the STATE table or STATE file on Target.
- Session Failure, rollbacks both changed data and tokens from the Target.
- When the Session completes, the tokens are stored in the physical file on the PowerCenter.
- During the Recovery, PWX reconciles the restart tokens in the physical file with Target State table or file.
How to create Restart Tokens?

• Use the token file listed below, to set the current restart time to all the tables

```sql
-- Restart Tokens for all tables
RESTART1=CURRENT_RESTART
RESTART2=CURRENT_RESTART
```

• Generate tokens from PWX Navigator by executing the SQL “SELECT CURRENT_RESTART ”

```sql
-- Restart Tokens for all tables
RESTART1=200812260735140000000010000000000000569C072DA70000000100000000
RESTART2=444144535430303120081226073437
```

• Create a wrapper script using DTLUAPPL utility to generate specific token values for each table.

```bash
DTLUAPPL > token.txt
```
Sample Token file

```plaintext
<!-- Restart Token for the Table: reqrmnt_UWW_REQUIREMENT -->
EXTNGRP.reqrmnt_UWW_REQUIREMENT=20080819185540000000400000FFFF000000000000000000000000000
EXTNGRP.reqrmnt_UWW_REQUIREMENT=444144535130313220080819185538

<!-- Restart Tokens for the Table: client_UWW_CLIENT_POLICY -->
EXTNGRP.client_UWW_CLIENT_POLICY=20080819185540000000400000FFFF000000000000000000000000000
EXTNGRP.client_UWW_CLIENT_POLICY=444144535130313220080819185538

<!-- Restart Tokens for the Table: activity_UWW_ACTIVITY_PL -->
EXTNGRP.activity_UWW_ACTIVITY_PL=20080819185540000000400000FFFF000000000000000000000000000
EXTNGRP.activity_UWW_ACTIVITY_PL=444144535130313220080819185538
```

• The PWX token file will always start with the symbol “<- -”
• The token information is specified in pair such as Restart1 and Restart2.
• Restart1 is known as Sequence token and Restart2 is known as Restart token.
• The Navigator Row test output includes these token values
  - `DTL_CAPXRESTART1`: Sequence Token (Timestamp + Position info)
  - `DTL_CAPXRESTART2`: Restart Token (Unique Database id + Timestamp)
PowerExchange CDC Restart Options

**Initial start**
Start the CDC session at a point in the change stream where the source and target are in a consistent state.

**Cold start**
The PWX reads only the tokens stored in the token files and will not check target STATE tables or files and sends token details stored in Restart token file to PWX. You can update restart token file to add or remove sources, when you cold start.

**Warm start**
The PWX compares the restart info in the token files with any info that exists in the STAT tables and creates new restart tokens with the reconciled restart tokens.

**Recovery**
The PWX reads the tokens from STATS tables located on Target and rereads the change stream and performs recovery processing.
PowerExchange CDC Recovery Strategies

Network glitches and Non Fatal outages
• Restart - PWX automatically restarts from the last check point

Check Point too old
• Sometimes, you might notice that the Restart tokens are too old and restarting from that point might cause huge number of rows to be reprocessed.
• In order to avoid this situation, always have a job to run DTLUAPPL utility to generate tokens in an archive log file periodically. Use the most recent Tokens from DTLUAPPL log file to restart the failed session.

Missing Archive Log files
• Restore the Target tables from Source Database backup, update the token files with CURRENT_RESTART and cold start.
• Go for PWX Logger option, which saves archive logs in the PowerCenter.
Utilities for CDC Implementation

### PWX Utilities

- **DTLINFO** - to verify the installation and to display the version, release and build level details.
- **DTLREXE** - to check basic connectivity to a Listener running on remote platforms.
- **DTLURDMO** - to migrate extraction maps from one environment to another
  - REG_COPY - copies Registrations
  - XM_COPY - copies Extraction Maps
- **DTLUAPPL** - to generate or modify Restart tokens.
- **DTLUCUDB** - DB2/UDB utility
  - DBINFO - displays database info
  - SNAPSHOT - initializes the capture catalog
  - LOGPRT - prints UDB Log file

### Command Syntax

- **DTLINFO LSTQA01**
- **DTLREXE PROC=PING LOC=INFA2PROD UID=xxxxxx PWD=xxxxxx**
- **DTLURDMO urdmo.txt > dmo.log**
- **DTLUAPPL uappl.txt > appl.log**
- **DTLUCUDB DBINFO DB=dadst01 UID=xxxx PWD=xxxxxx**
PowerExchange CDC Installation

**Requirements and Prerequisites**
- UDB must be at Version 8 FIXPAC 11 or greater.
- Data Capture Changes must be enabled for any table which is to be registered for capture.

**Restrictions and Caveats**
- LOB or XML data types.
- Abstract data types such as Stacks, Queues, Lists.

**Installation steps**
- Update PowerCenter profile with PWX Variables such as PWX_HOME, PATH, LD_LIBRARY_PATH etc.
- Install PWX on Integration Service Machine.
- Install PWX Navigator on the Client Machine.
- Register Listeners and add PowerExchange nodes in dbmover.cfg configuration file.
How to Implement PWX CDC?

Prep Tasks for Implementation

- Have database User-id for Source database with SYSADM or DBADM privileges handy.
- Configure the Source and check whether Archive logging is on.
- Decide how long to retain Archive Logs, if the PWX Logger option is not chosen.
- Enable Data Capture Changes for the Source tables using the alter table DDL.
- Get the DDL for CCATALOG table from the installation manual and create the catalog table on the Source database.
- Check if the Recovery and State tables are generated on the Target database and make sure the user-id has privileges to create these tables on Target database.
How to implement PWX CDC?

- Before executing any PowerExchange commands, make sure the DB2CODEPAGE is set to 1208 and DB2NOEXITLIST is ON.
- You might optionally want to update CATALOG table with the new default Start time using DTLUCUDB UPTDRP utility.
- Before starting the CDC sessions, initialize CATALOG table on the Source database using DTLUCUDB SNAPSHOT utility.
- Activate Listener using DTLLST and check the connectivity using DTLREXE PING utility.
- You can migrate extraction maps using DTLURDMO utility.
- Perform Database Row test from Navigator to test the Source and Target Connectivity.
- Perform Database Row test using Real-Time mode (CAPXRT) and check the data in real time to make sure the Extraction Maps pull the correct data.
How to Implement PWX CDC? (3)

Materialize Target tables
- Have DBA run the pre-replication Source database backup.
- Restore Target from source database backup.
- Check the tables, permissions and compare row counts after the Target restore.
- Make sure the Source and Target are in a consistent state.

Schedule CDC Workflows
- Determine the time to kick-off the CDC workflows.
- Schedule Workflow to run continuously

Validate, Verify and Monitor
- Check the CDC Detail log file and session performance details to make sure the CDC is working as intended.
- Check CPU, I/O stats and resource usage on Source, Target and the PowerCenter servers.
Tips & Tricks

• Use CDC Group Source feature by adding multiple sources in a mapping, so that all the change data from these sources can be captured in a single read.
• Use PWX Generated columns in the Mappings, for example you can use CAPXTIMESTAMP in the Source Qualifier to retrieve the changes based on a specific time.
• The PWX Generated Column-Level Change indicators may be used to select and process only the changed Columns.
• In order to avoid failures due to rejects, load all the changed data either as Inserts or Updates based on whether the record exists on the Target and the record action type using the PWX generated CAPXACTION column value in the change record.
• You can trickle-feed Dynamic Lookup Caches from CDC sources for warehousing and various reporting applications.
Tips & Tricks

• CDC session freezes, when the Listener is running out of buffer memory. This is mainly due to the huge UOW from the Source. The workaround is to lower the UOW count and the maximum number of rows per commit.
• The PWX writes the latest tokens into the Restart token file, only when it is complete successfully. Hence stop and restart the CDC sessions periodically to save the token files.
• CDC sessions can be stopped and restarted automatically from the continuously running workflows.
• You can reset the Idle time value to 0, so that the session can be made to stop, when the reader reaches the end of log. This will force the session to stop and save the tokens.
• You can create a job in Crontab, to monitor the Listener and to generate tokens in a file using the DTLUAPPL utility.
References

1. Informatica How-To Library
2. Informatica Knowledge Base
3. PowerExchange DB2 Adapter Guide
4. PowerExchange E-learning
5. PowerExchange Getting Started Guide
6. PowerExchange Installation Guide
7. PowerExchange Message Reference
8. PowerExchange Migration Guide
10. PowerExchange Reference
11. PowerExchange Software Requirements
13. PowerExchange Utilities Guide
14. Velocity Best Practices

Questions? Contact us at
Allen.Bailey@us.ing.com
Victor.Kennedy@us.ing.com