IBM InfoSphere Data Replication (IIDR)

June 2013

Jayanti Mahapatra

IBM Data Replication
mahapatr@us.ibm.com

Agenda

- *Why Replication?*
- *InfoSphere Data Replication*
- *A technical review of IIDR's Q Replication*
- *A technical update of IIDR's Q Replication (V10)*
- *Q Replication Customer use cases*
- *Performance improvements*
- *Backup slides*

Why Replication ?

- **Continuous Availability and Disaster Recovery**
  - *Failover Capability for Disaster Recovery (DR)*
  - *Workload Distribution (read/write at each site)*
  - *Requires awareness of replication latency, conflict avoidance and resolution policies*
- **Live Reporting**
  - *Offload the OLTP server, for Reporting or Analytics over (almost) live data*
- **Data Distribution and Consolidation**
  - *Multiple locations (e.g., hundreds of Points of Sales, or Business Branches)*
  - *Often bidirectional, assumption of no conflicts (e.g., rollup sales, distribute price lists)*
- **Maintenance or Migrations *without application outage***
  - *Hardware or Software upgrades*
- **Feeding the Warehouse**
  - *DB2 → [Staging →] ETL*
  - *Row-Level transformations (outside of ETL)*

IBM InfoSphere Data Replication (IIDR)

The InfoSphere Data Replication(IIDR) product offers 3 technologies:

1. SQL Replication (aka DB2 Data Propagator - since 1994)
   - *Table updates are captured by reading the logs for DB2 sources, on z/OS, distributed, and iSeries*
   - *Captured changes are stored in relational tables, the Apply process fetches them via a database connection*
   - *Oracle, Sybase, SQL Server, Informix Teradata targets updated via federation*

2. Q Replication and Q Event Publishing (since 2004)
   - *Table updates are captured by reading the logs for DB2 on z/OS , DB2 distributed and Oracle*
   - *Captured changes are delivered via WebSphere MQSeries*
   - *Parallel Q Apply provides high performance*
   - *Oracle, Sybase, SQL Server, Informix Teradata targets updated via federation*

3. CDC

*The Capture, Administration and Monitoring programs are common to SQL and Q replication

The IBM InfoSphere Data Replication (IIDR)

- ***Software-based Replication solution***
  - Low latency (immediate failover)
  - Unlimited distances, un-like source/target systems (hardware and software)
  - Logical Replication: filters on tables, columns, transactions, operations

- ***Database Replication solution***
  - Changes captured asynchronously from the db recovery log with minimum impact to source
  - Unit of replication is a source transaction, preserves transaction consistency
  - Database conflicts are managed (detection, resolution)

- ***High Availability solution***
  - Active/Active scenarios: Database is available for read/write at each site, allowing workload distribution
  - Flexible recovery procedures during outages (system, network, etc.): Capture and Apply programs asynchronous of one another, data staged in queues, etc.

Technical review of IIDR's Q Replication

Why Q Replication (1)

- ***Performance and Scalability***
  - *Q Replication is capable of replicating 1000s of tables, 1000s of changes per second over distances of 1000s of miles with sub-second end-to-end latency (DB2 commit at source to DB2 commit at the target)*
  - *With resiliency. Tolerating hardware, software, and network failures.*

- ***Continuous Operations***
  - *Add/remove tables to an active replication configuration*
  - *Add new columns to a table already being replicated*
  - *Alter column set data type to an active replication configuration for Q rep*
  - *Take action when some DB2 Utilities are detected in the log for source tables (LOAD, REORG)*
  - *Perform Target Table LOADs in parallel*
  - *Tolerate Target Table Maintenance (e.g., REORG of a table at the target)*

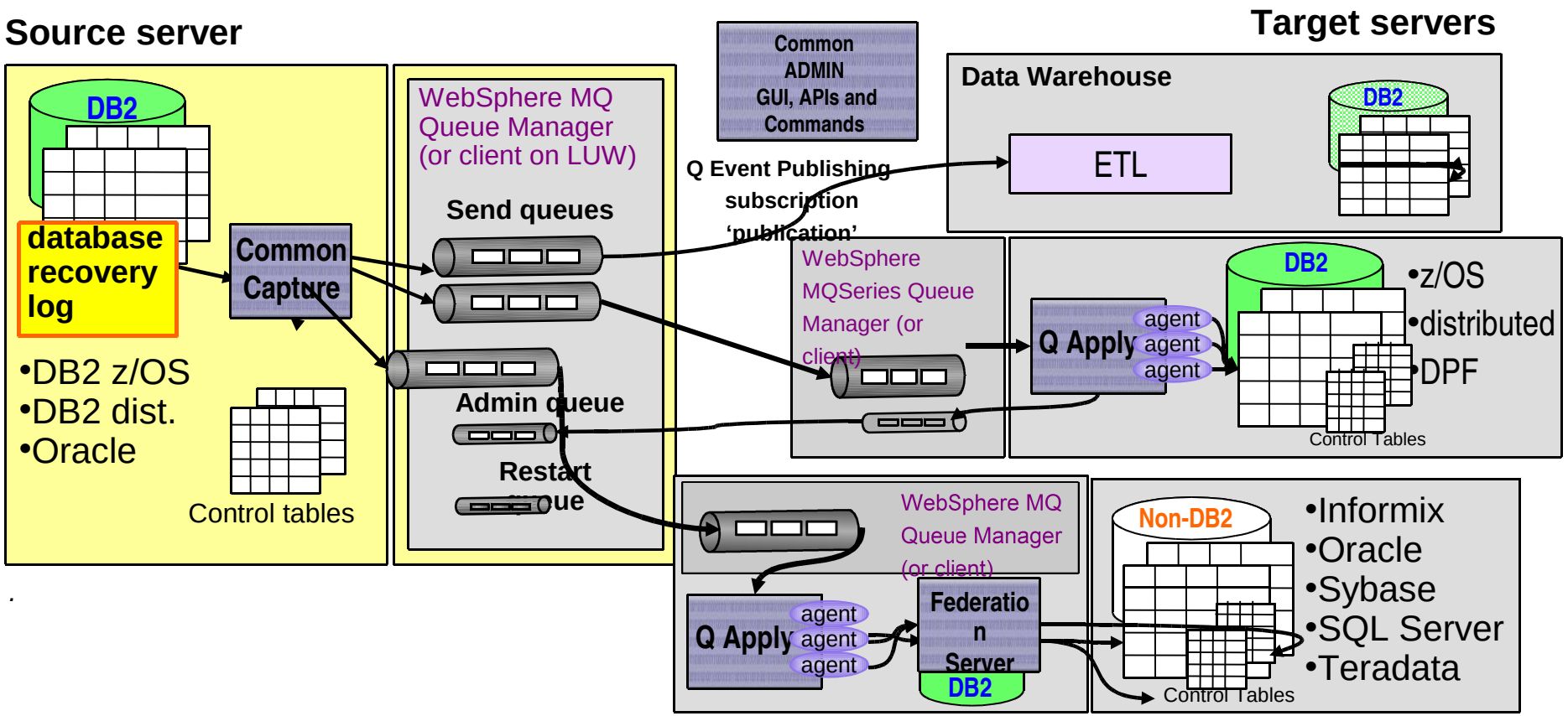
  <u>Without interruption of service</u> (no application outage, no replication suspend)

Why Q Replication (2)

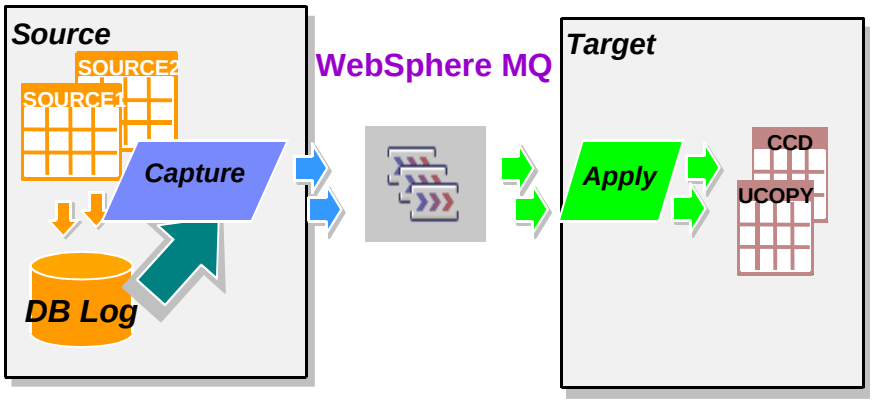
- ***Monitoring and Troubleshooting (Tooling)***
  - *Historical and real-time information about operation and performance in easy to query DB2 tables*
  - *Web-based Data Replication Dashboard for monitoring and trouble-shooting with powerful reporting facilities*
  - *Powerful Table Difference Utility (ASNTDIFF)*
  - *Comprehensive set of online documentation, samples, classes and tutorials*
- ***Strategic Solution for GDPS Active/Active***
  - *Development Focus on Active/Active, Manageability, Usability, etc.*
- ***Large Customer Base***
  - *Thousands of customers for both Q and SQL Replication*

Q Replication and Q Event Publishing

- ***High-throughput. Low-latency.***
  - *Typically tens of thousands of rows/second over 1000s of miles, with sub-second latency*
  - *Multi threaded QApply program. Can handle Referential Integrity and Unique constraints at the target, reorder transactions, if and needed.*
- **Staging and transport over MQSeries**
  - *Captured database transactions published in MQSeries transaction (as per user specified commit_interval)*
  - *Tolerates site down, changes accumulate in persistent queues, very fast recovery*
  - *No 2PC commit with MQSeries in either the Capture or Apply program*
- **Apply can perform initial table loads, in parallel, without need to suspend user workload or replication**



Q Replication Overview



Q Capture program

Reads changed data from the log

Stages changed data in MQ queues

MQ Series

Delivers data to system where Apply program runs

Limited staging at the source

Q Replication supports all supported levels of MQ

Q Apply program

Pulls data from MQ queues and applies to target tables

Never connects to the source for changed data

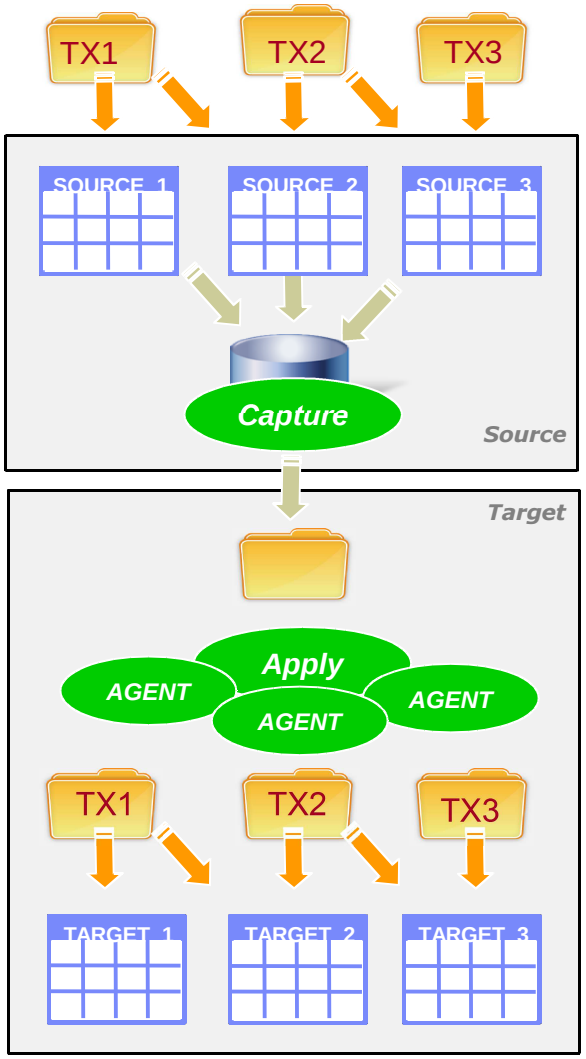
Parallel Apply technology: for high performance

Administration Tools common with SQL Replication

High Performance Q Replication

- **Parallel Apply**: Non-dependent transactions applied and committed in parallel at target
  - Can significantly reduce latency of target data
  - Performance can scale almost linearly

- Serial Apply mode: Dependent transactions applied in source commit order
  - Q Apply checks for constraints: key, unique, RI.



Q Replication for High Availability

- Allows read or write activity at two sites (bidirectional replication)
- Choice of conflict detection and resolution options
- Fast switchover with transaction-consistent data (scheduled or non-scheduled)
- Platform and distance flexibility between sites
  - *Hardware, OS level, DB level, application level can all be different*
  - *Can be geographically dispersed (unlimited distance)*
- Two approaches with bidirectional replication
  - *Primary site with a Stand-by*
  - *Two sites active for updates (Active-Active)*

Focus: Making data available while allowing for planned outages and can be combined with a DR solution for unplanned outages

Technical Update - IIDR Q Replication
IIDR V10 – Arch_level 100z (z) and 1001 (luw)

Q Replication V10 Features

1) Replicate ALTER TABLE ALTER COLUMN SET DATA TYPE
2) Auto-handle ALTER ADD column at the source
3) Restart at a queue level (Q Capture override restart option)
4) Tolerate DB2 for z/OS and MQ for z/OS outages
5) Detect and propagate source table loads (DB2 utilities)
6) Detect and handle DB2 maintenance on target tables (Q Apply option to spill data to queues)
7) Control stop of Q Capture (planned outage support)
8) Q Capture Warning for large transactions
9) Parallel ASNTDIFF
10) Q Capture multiple DB2 transactions in one MQ message
11) Q Capture 'ignore trans' wild card support

Continuous Availability : Focus: No interruption to the business - Goal: 100% availability

1) ALTER TABLE ALTER COLUMN SET DATA TYPE

- Detect ALTER TABLE ALTER COLUMN SET DATA TYPE from the DB2 log
- Capture sends the DDL operation to Q Apply to replay at the target
- Q Apply maps name between source and target as long as the mapped column is not mapped to an expression
- If an expression, user must change the expression/underlying target table

2) Auto-Handle ALTER ADD COLUMN at the source

ALTER ADD COLUMN is detected for source tables from the DB2 log.

ALTER ADD COLUMN is replicated if the option is set for the subscription (REPL_ADDCOL).

New columns will be automatically added to the target table if they do not already exist at target (and Q subscription metadata for that table will be updated).

3) Q Capture LSN restart at Queue level

Q Capture supports multiple consistency groups.

A consistency group is a workload of tables replicated from one (source) server to another (target) server:

- Each consistency group is associated with a Replication queue
- Q Capture always maintains a min and max log sequence values for each consistency group/Replication queue
- In case of error in one queue:
  - Q Capture doesn't stop publishing to the other queues
  - When the error is corrected, Q Capture can restart from where it last read the log (unless a specified point in the log is provided) for the queue in error with minimal impact to the other queues

4) Tolerate DB2 for Z/OS and MQ for z/OS Outages

- Q Capture and Q Apply detect when DB2 or MQ become unavailable (may not be started before Replication starts or stopped while Replication is running)
- If the appropriate option is provided (TERM=N), Q Capture and Q Apply will attempt to reconnect to DB2 and MQ until they are available
- If that option is not provided, Q Capture or Q Apply will come down

5) Detect and Propagate Source table Utilities

- Q Capture detects the following events from DB2 log at the source (**for IBM utilities only):**
  - *LOAD REPLACE*
  - *LOAD RESUME SHRLEVEL NONE*
  - *REORG DISCARD*
  - *RECOVER PIT*
  - *CHECK DATA DELETE YES LOG NO*
  - *EXCHANGE DATA ON CLONE TABLESPACE*

- Q Capture always issues a warning when it detects one of the above events
- If option is provided for Q Capture to propagate the load, Q Capture sends a schema message to Q Apply which in turns loads the the target table
- Whether the load is propagated or not, the subscription remains active

6) Detect and Handle DB2 maintenance on target tables

- Q Apply **detects** DB2 maintenance (-904 DB2 SQL error messages)

- Q Apply **handles** the error based on the target table 'error action' as follows:
  - Deactivates the subscription
  - Stops the queue
  - Comes down
  - Or puts the target table subscription in 'spill' mode:
    - Data is spilled to a temporary spill queue while the table is unavailable
    - A persistent spill queue is automatically created as per the model definition
    - The subscription is placed in 'S' or spilled state
    - User needs to issue a RESUMESUB command after the error condition is cleared to alert Q Apply to process data from the spill queue
    - After Q Apply clears up the backlog of changes, it deletes the spill queue and places back the subscription back in active mode

7) Q Capture Controlled STOP

- Planned Maintenance support
- 2 options are available for synchronization between Q Capture and Q Apply:
  - Q Capture stops after data has been sent to the target MQ (checks that the transmit queue is empty – remote queue only)
  - Q Capture stops after data has been applied at the target (waits for Q Apply acknowledgement)
- Supported via command, signal, MODIFY command (z)
- Can be specified at queue level and status reported through messages to the console or job

8) Q Capture Warning for large transactions

WARNTXSZ=nn :

- Threshold value is in Meg
- Every time the threshold value is exceeded, Q Capture issues an ASN0664W warning
- If WARNTXSZ = 10 and transaction size is 30, 3 warnings are issued by Q Capture (one for 10M, one for 20M and one for 30M).

This function is also available in V97

9) The Highly Parallel ASNTDIFF

Use the highly parallel ASNTDIFF to compare the contents of source and target replicated tables

Highly Parallel and highly optimized

- Parallelized processing (partitioning, detecting the differences, reporting)
- Minimized traffic over the network (only sends data needed)
- Exploitation of computational capabilities of systems (checksum calculation)
- Using Stored Procedure (ASNTDSP) which runs at source and target systems

10) Q Capture Multiple DB2 transactions in one MQ message

- **TRANS_BATCH_SZ parameter:**
  - *Maximum number of source database transactions to group in one MQ message*
  - *More efficient transport across the network and CPU reduction*
  - *Especially efficient for workloads where each transaction modifies only a few rows*

Technical Update - IIDR Q Replication
IIDR V10 – Arch_level 1001 (luw only)

DB2 for Linux, UNIX and Windows only

Schema-level Subscription Support (DB2 for Linux, Unix, Windows V10)

- Automatically replicate some DDL operations using **schema**-level subscription:
  - *CREATE TABLE*
  - *DROP TABLE*
  - *ALTER TABLE ADD COLUMN*
  - *ALTER TABLE ALTER COLUMN SET DATA TYPE*
- Greatly reduce the need for administrator intervention during replication activity
- Use a single command to set up replication on an entire database, a group of schemas or a single schema

V10 Update: DB2 Temporal Support in Q Replication

- Temporal table support in unidirectional and bidirectional replication
- You create a Q subscription for the base temporal table and a separate Q subscription for the associated history table if you are using system-period temporal tables or bi-temporal tables.
- A built-in stored procedure prompts DB2 to take the following actions at the target database:
  - *Not write rows to the target history table for update or delete operations. Instead, these row changes are replicated from the source history table.*
  - *Allow the insertion of values from the source temporal table for row-begin, row-end, and transaction-start-ID columns even though these columns are defined as GENERATED ALWAYS.*
  - *On Linux, UNIX and Windows, allow TRUNCATE statements on system-period temporal tables to support LOAD REPLACE operations on these tables.*

Technical Update and Performance Improvements - IIDR Q Replication V10 – Arch_level 1001 (z)

What's New Since Data Replication V10

APAR PM84946 (arch_level 1001) SQL and Q Replication includes:

- Throughput Increase, Latency Reduction
  (Parallel send queues, Message persistence option)
- New Conflict rules for Uni-directional subscripton
- Temporal table support
- MQ V7 APAR PM63802 (MQ Read ahead function) helps replication performance

* **Replication APAR PM84946 will require replication control table migration**

* **DB2 HIPER APAR PM84864 recommended**

V10 Update: Q Replication Throughput and Latency Performance Improvements (Parallel Send Queues)

Scenario:

High rate of small transactions.

QMGR transmission queue at source becomes a bottleneck and slows down Q Capture.

Q Apply is not kept busy and end-end latency dramatically increases.

Symptoms:

a. Growing transmission queue depth at Q Capture side reported in IBMQREP_CAPQMON XMITQDEPTH value (XMITQDEPTH added in APAR PM75119)

b. High MQPUT_TIME relative to the MQ_MESSAGES as reported in IBMQREP_CAPQMON for that queue

c. Growing QLATENCY but low QDEPTH in IBMQREP_APPLYMON for that queue

V10 Update: Q Replication Throughput and Latency Performance Improvements (Parallel Send Queues)

Solution:

Assign **multiple (two recommended) send queues** to a single receive queue.

Up to **80% throughput increases** and **90% decrease in end-end_latency** observed in the lab (using 2 send queues). However, also, up to **12% CPU increase in Q Capture CPU** has also been observed for this scenario.

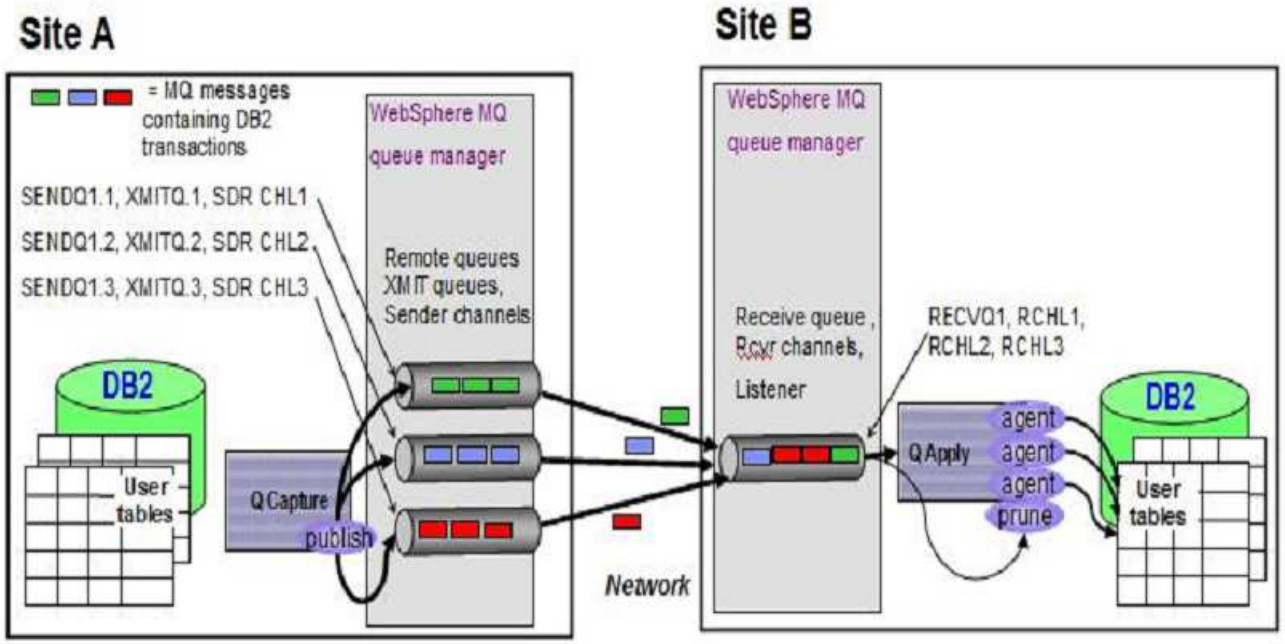
Multiple transmission queues and sender channels for a single Replication queue map allow the source QMGR to keep up with the transaction rate. Use:

- NUM_PARALLEL_SENDQS > 1 in IBMQREP_SENDQUEUES table
- IBMQREP_RECVQUEUES PARALLEL_SENDQS='Y' .

This option is valid for **z/OS only** and requires Q Replication APAR PM84946

V10 Update: Parallel Send Queues At-a-Glance (z/OS)

- Multiple send queues can link to the same receive queue
- Up to 80% Performance improvement and 90% decrease in end-to-end latency seen in lab tests using 2 send queues.



V10 Update: Q Replication Throughput Increase and CPU Cost Reduction (Non-Persistent MQ Messages)

Scenario:

Some Applications require higher throughput and lower CPU cost.

These (non-critical) applications are willing to trade-off recoverability for better performance.

V10 Update: Q Replication Throughput Increase and CPU Cost Reduction (Non-Persistent MQ Messages)

Solution:

Use non-persistent MQ messages as they are not logged in MQ.
Up to **10 % to 20% CPU reduction** and **80% throughput (rows/second) increase** have been observed.

New option in V10 honors the MQ queue DEFPSIST persistence setting in MQ: use MSG_PERSISTENCE=D instance level/Q Capture parameter

Recovery procedure: Use Q Capture OVERRIDE_RESTART option to use the LSN and MAXCMTSEQ restart information for the queue (see next slides)

The Data Replication Dashboard Recovery Advisor also steps you through Recovery procedure steps to recover lost MQ messages (see Lab)

V10 Update: Use Replication Dashboard or Manual Procedure to Recover from lost MQ Messages

To recover lost MQ Messages:

- Use **Data Replication Dashboard Recovery Advisor** or
- Follow the manual procedure
  1) Derive the 'LSN restart' values for the appropriate queue(s) by using the **Q Apply MAXCMTSEQ** as an upper bound in a SQL query against the IBMQREP_CAPQMON to find a safe LSN value to start from for the queue(s)
  2) Update the restart file with information from above. Restart file is a z/OS data set: *capture_path.capture_server.capture_schema.QCAP.QRESTART*
  3) Use V10 Q Capture 'LSN restart' option at **Queue Level** to warm start Q Capture using the restart file information, ie back from a point in the log that will recapture the missing MQ messages for the relevant queue(s), for example, queue(s) with non-persistent MQ messages that were lost

Review: Overriding Q Capture Warm Start LSN Values

- **Before V10:**

  - LSN and MAXCMTSEQ parameters at Q Capture **program level** only
  - LSN: start reading DB2 log from this LRSN value
  - MAXCMTSEQ: publish source data committed after this log sequence value

- **In V10:**

  - LSN and MAXCMTSEQ override values available at **queue level** option
  - Q Capture OVERRIDE_RESTARTQ=Y parm allows restart from different LSN values for each queue in the restart file (Q Capture uses minimum value from its restart file and from its restart message information):

```
//QASN1CAP  EXEC PGM=ASNQCAP,REGION=0M,TIME=NOLIMIT,
// PARM=/CAPTURE_SERVER=DSNT CAPTURE_SCHEMA=DEMO??U1
//          OVERRIDE_RESTARTQ=Y'
```

  - Q Capture temporarily stops publishing to active queues if other queues restarted and catch up (Q Capture still one log read thread)
  - Q Capture updates the restart file and job log when stopping with the current LSN values for each queue

V10 Update: DB2 Temporal Support in Q Replication(z/OS)

Scenario:

User replicates DB2 base temporal tables and their history tables to a target server

Solution:

- Requires DB2 APAR PM61811 and Replication APAR PM84946
- Create the following Stored Procedure on z (SDSNSAMP(DSNTESR) :

```
CREATE PROCEDURE  SYSPROC.SET_MAINT_MODE_RECORD_NO_TEMPORALHISTORY ()
 EXTERNAL NAME DSNTPRC1   LANGUAGE C ;
```

- Modify configuration options for DB2 admin routine DSNTIPRL to specify list of userids to be granted 'execute' access on the procedure by job DSNTIJRT:

```
PACKAGE OWNER ===>
SET_MAINT_MODE_RECORD_NO_TEMPORALHISTORY
GRANT EXECUTE ===>  Apply task userid
```

V10 Update: New Conflict Rules for Unidirectional Subscriptions

Scenario:

User defines bidirectional replication between two sites and relies on a pair of 'unidir' Q subscriptions to avoid the limitation of the Q Replication 'bidi' subscription (for ex, unlike table structures between the two sites).

The user also needs the more generic **conflict rules of 'C' or 'A'** available with 'bidi' Q subscriptions to detect changes to non-key columns.

Solution:

Allow 'unidir' Q subscriptions to use conflict rules 'C' (all key columns plus any changed non-key columns) and 'A' (all columns) when detecting conflict. The existing behavior for 'unidir' Q subscriptions is to only support the 'K' or key columns only rule when detecting conflicts.

V10 Update: Q Apply Throughput Increase ('MQ Read-Ahead' function)

Scenario:

Q Replication latency increased due to a workload spike/changed data built up in queues (temporary unavailability of the target database) so the Q Apply program and MQ channel agents can't pull messages from memory and revert to disk.

Solution:

Enable Q Apply/MQ channel agents to read messages from memory (the MQ '**read-ahead**' enhancement introduced in WebSphere V7 APAR PM63802)

Up to a **55% Q Apply throughput increase** has been observed with improvements more dramatic for **smaller message sizes**

No change to the Q Replication environment required to take advantage of the MQ feature

V10 Update: Q Apply CPU Reduction during Pruning ('MQGet for Delete' function)

Scenario:

Q Replication latency is impacted by the MQ pruning cost in Q Apply

Solution:

Enable Q Apply pruning logic to use "**MQGET for Delete" function** - requires MQ V7 PM63802 or higher

In controlled environment, **on start up of Q Apply 20% CPU reduction** has been observed for Q Apply **and 70% CPU reduction** in MQ only when there are many messages in the queue that needs to be pruned.

With this new function, MQ will not retrieve the MQMD part of the message which results in fewer MQ IO's and Q Apply will request destructive reads of a message (with buffer size 0) optimizing the pruning process

Interesting customer use cases

Case Study: Banco do Brasil (DB2 z/OS)

Challenge

- Excessive load on production system due to combination of an increasing OLTP workload and query reporting applications
- Two SYSPLEXES with unbalanced workloads
- Need for a workload balance solution with low impact on existing applications
- Critical and non-critical applications sharing state of the art infra-structure resources

Solution

- InfoSphere Replication Server's Q Replication
- Use the lightly loaded secondary system to offload query reporting applications and to separate critical and non-critical applications in different DB2 data sharings groups.
- Replicate thousands of tables to the secondary system using uni and bidirectional replication to integrate the application data.

Business Benefits

- Improved application and data availability
- Reduction of unplanned application outage
- Improved SLA with Line of Business
- Focus investments in infra-structure for critical business applications

Cost Benefits

- Better sysplex utilization
- No need for application redevelopment, minimum changes required

Banco do Brasil (DB2 z/OS)

- Scenario: Workload balance
- DB2 z/OS to DB2 z/OS V10 CM
- Configurations: bidi, uni-directional
- Number of queues: 80
- 4 different DB2 data sharing groups involved
- 6 Q capture
- 7 Q apply
- Number of subscriptions: Over 2000
- Replicate over 500 millions rows/day at peak period

Case Study: UBIS(DB2 z/OS)

Challenge

- Top financial services firm in Europe with growth through acquisition strategy.
- Very heterogeneous data environment with multiple operating system platforms and databases.
- Regulations and internal mandates required constant synchronization between 70 different locations of DB2 alone.
- Remote access from many applications let to organizational confusion

Solution

- InfoSphere Replication Server's Q Replication to provide constant availability of synchronized data between all systems
- Implement 18 Q Capture components and 33 Apply components to replicate over 5000 tables between many systems

Benefits

Unicredit now has a single view of its information assets, regardless of where the information resides.

Reduced development costs through consolidation of data from many locations

Achieved high-volume, low-latency data replication with managed conflict detection and resolution

Minimized latency by maximizing parallelism of the Q Apply program

Q Replication's data transformation option eliminated need for custom applications.

Unicredit (DB2 z/OS)

- Scenarios: Live Reporting, High-Availability
- DB2 z/OS to DB2 z/OS V10 NFM
- Configurations: bidi, uni-directional
- Number of queues: 123
- 70 different z systems
- 18 Q capture
- 33 Q apply
- Number of tables: Over 5000
- Special features used: Q Apply expressions

Backup Slides

Q Replication Tooling

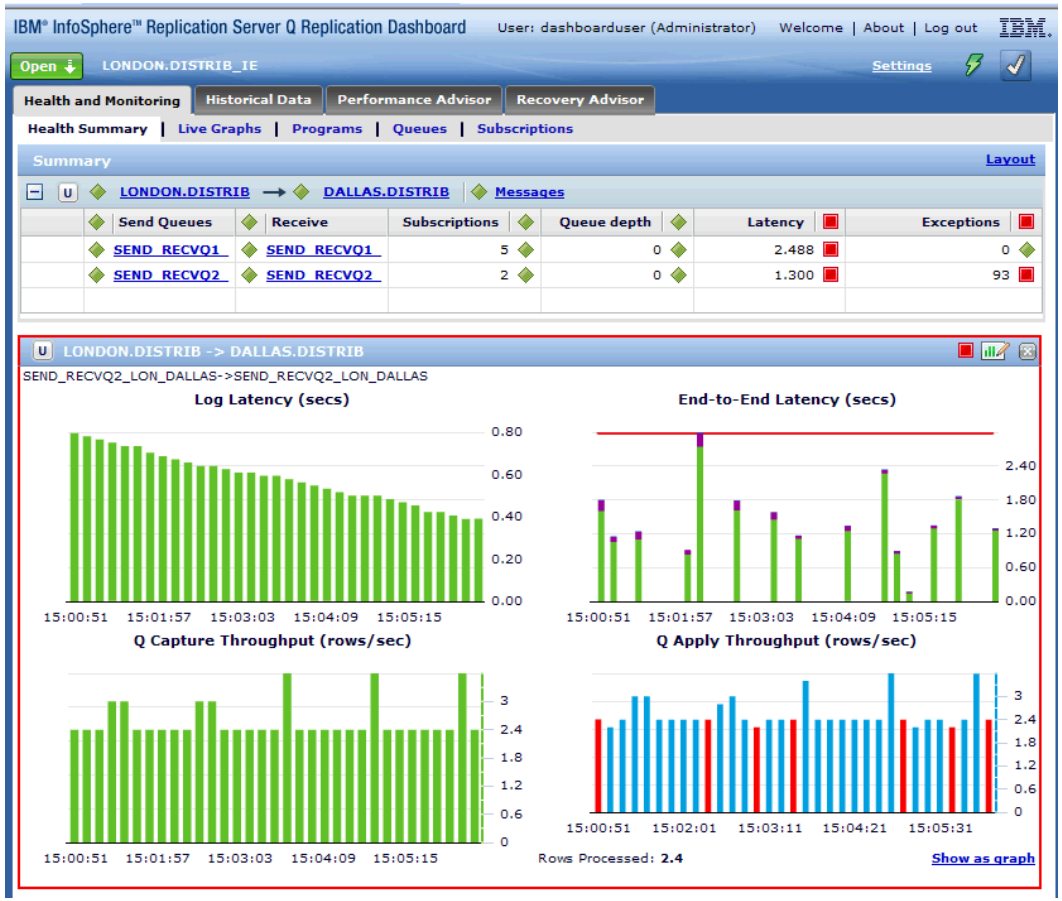
1. Replication Center and command-line processor ASNCLP for **administering** Replication definitions - Available as a free download in the IBM Data Server Client
2. Data Replication Dashboard for **live monitoring, trouble-shooting, operating and reporting –** with option to send email - Available as a free download from the IBM support website
3. **Table Compare and repair** utilities ASNTDIFF
4. Utility for **background monitoring** ASNMON alerts for problem conditions with option to send email/page
5. Program (modify) commands or user signal for **operating**

Command line admin (ASNCLP) runs natively on z/OS

- Can use Job Control Language (JCL) to run the ASNCLP command-line program on z/OS
- Enables you to generate SQL scripts for creating and changing replication objects without the need for a remote Linux, UNIX, or Windows system to connect to the servers
- ASNCLP command scripts are defined as input data sets in the JCL (INMAIN), and the generated SQL scripts are defined as output data sets
- ASNCLP uses the communications database in a DB2 for z/OS subsystem to connect to the source and target servers

The Data Replication Dashboard: Q Replication Monitoring

- **Live Monitoring**
- **Runs from Web Browser**
- **Multi-sites monitoring**
- **Troubleshooting**
- **Tuning and Reporting**
- **Alerts (email, visual)**



The Data Replication Dashboard

- **Performance Advisor**
  - *Analysis of Q Capture and Q Apply based on historical performance data*
  - *Prioritized tuning recommendations given to improve latency.*
- **Recovery Advisor**
  - *Aid the user in disaster recovery situations like loss of MQ message*
  - *Guide user through recovery process and generate scripts*
- **Deploy dashboard to existing WAS web server**
  - *In addition to pre-packaged embedded Web Server.*
- **Able to email generated reports**
- **Compact summary view to accommodate many data direction on one page. User can switch between 2 summary view layouts now**
- **Throughput graph in the latency report**
- **View and change table content for replicated source and target tables**

Staging Changed Data:

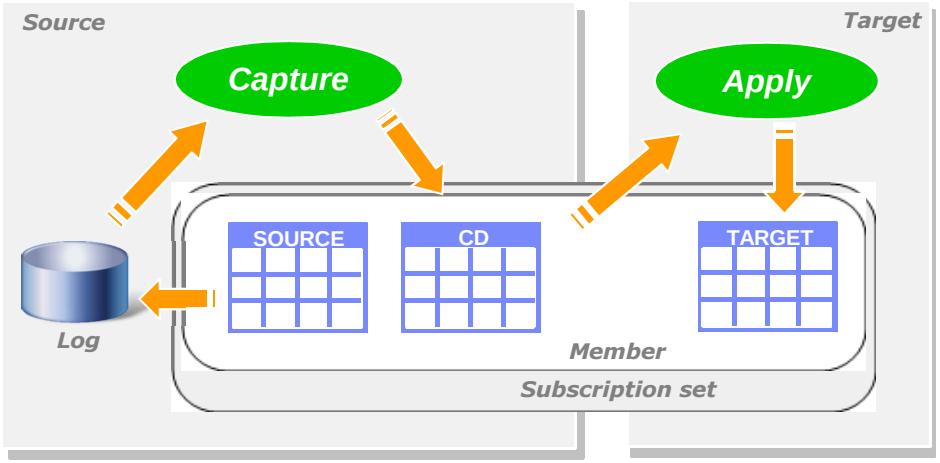
An Introduction to SQL Replication

SQL Replication Basics

SQL Capture program

Reads changed data from the log

Stages changed data in (CD) tables



SQL Apply program

Pulls changed data from CD tables

Replays transactions on target

DB client-server communications

Supports unidirectional and update anywhere

SQL-based

All meta-data in tables as well statistics and messages

Capture and Apply are database applications

Replication Center and Command line interface (ASNCLP)

Staging of Data in DB2 tables

An excellent model for Data Distribution and Consolidation

Scales to many targets with ease

supports occasionally connected targets

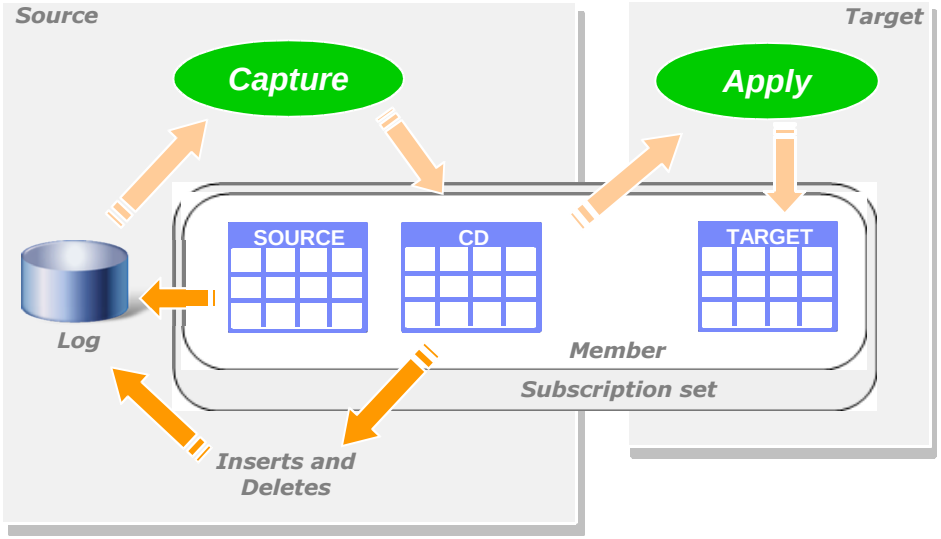
Takes full advantage of SQL

Allows for flexible subscription definitions (all in SQL)

Log Overhead

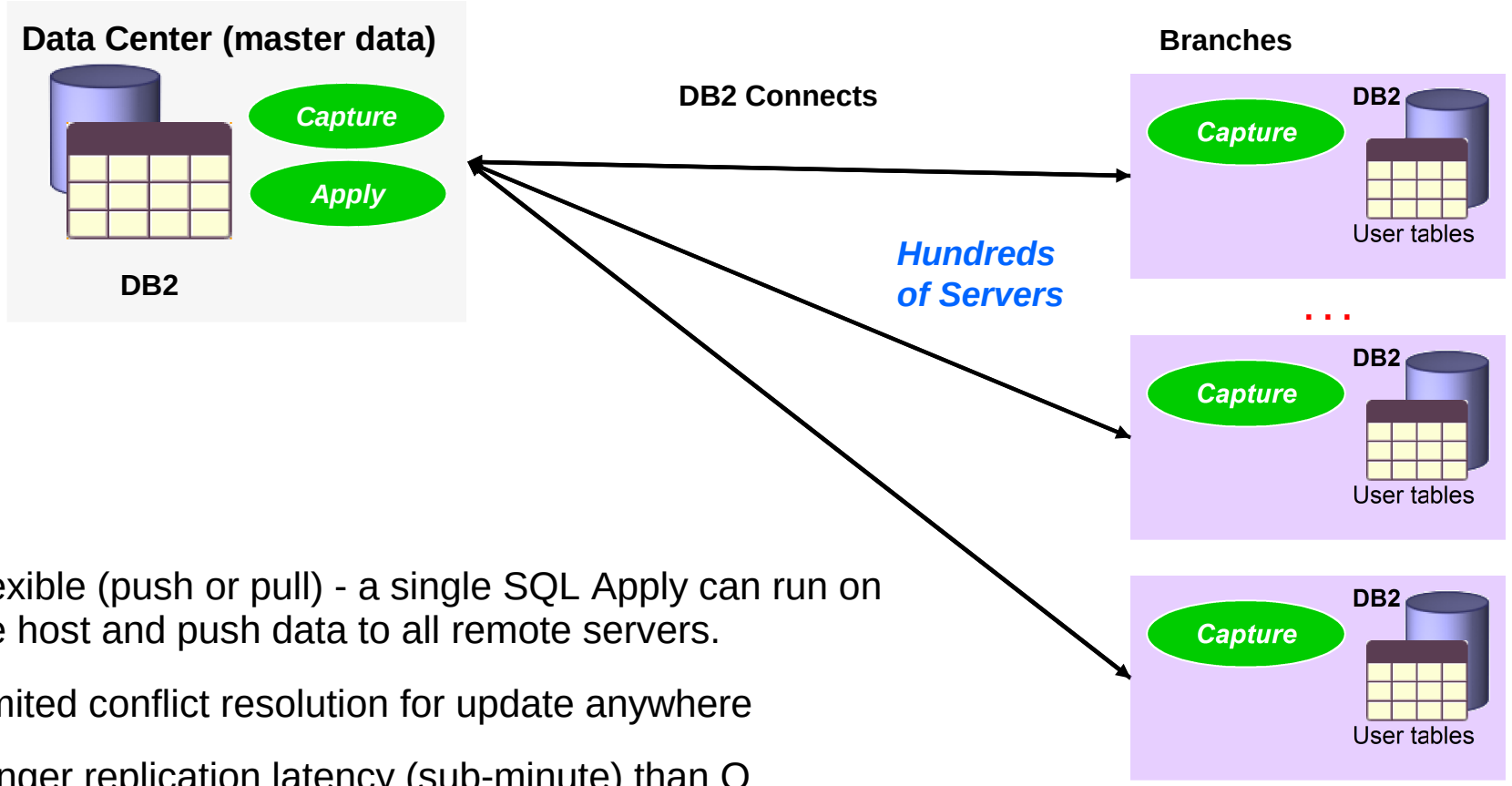
Inserts into CDs

Pruning of CDs (deletes)



DB2 z/OS V9 NFM or higher provides the alternative of non-logged table spaces → eliminates log overhead on CDs

Ex: Large multi-way consolidation/distribution scenario



- Flexible (push or pull) - a single SQL Apply can run on the host and push data to all remote servers.
- Limited conflict resolution for update anywhere
- Longer replication latency (sub-minute) than Q Replication (sub-second)

SQL Replication Enhancements

- Replicate ALTER TABLE ALTER COLUMN SET DATA TYPE
- Replicate ALTER ADD column
- SQL Apply APPLYMON table updates

ALTER TABLE ALTER COLUMN SET DATA TYPE

- **Detecting ALTER TABLE ALTER COLUMN SET DATA TYPE**
  (DB2 APAR PM05503 V9 PTF UK63236 and V10 PTF UK63237)
  - *SQL Capture automatically alter CD table (both before and after columns)*
  - *SQL Apply will not detect the DDL changes; User needs to alter the target*

Replicate ALTER ADD COLUMN

- **SQL Replication supports ALTER TABLE ADD COLUMN**
  - *Capture tolerates the new added column Users must still manually add the column to the Change Data and target table.*

SQL Apply: New Apply Monitor Table

- New Apply control table IBMSNAP_APPLYMON:

  Apply program's current state
  Current subscription set being processed
  Current member table being processed

- New Apply monitor thread
  writes status information to the new control table without affecting overall program performance