An Introduction to WebSphere Information Integrator Q Replication

Tridex       June 16, 2005

Beth Hamel
DB2 Replication Product Architect
hameleb@us.ibm.com
Agenda Topics

- The Basics of MQ Based Replication
- Publishing DB2 data to MQ in an XML format
- Application Examples of Replication and Publishing
Flexible scheduling, transformation, distribution
Typically used for business intelligence, distribution and consolidation, application integration
Why Create Another Replication Architecture?

- **Performance:** Combine high throughput with low latency

- **Capability:** Significantly improve multi-directional replication support

- **New function:** Event publishing, table difference utility

- **Manageability:** Reduce the number of replication objects to be defined and managed, ease the definition process with new Replication Center wizards
Q Replication Architecture

- Each message represents a transaction
- Highly parallel apply process
- Differentiated conflict detection and resolution
- Integrated infrastructure for replication and publishing
- Staged availability of heterogeneous support
Q Replication – Q Subscription Process

SOURCE

- SOURCE1
- SOURCE2
- DB2 Log
- Q Capture
- METADATA

ADMISTRATION

- Replication Center
- Replication Monitor

TARGET

- Q Apply Browser
- Apply Agent
- Apply Agent
- TGT1
- TGT2
- TGT3
- METADATA

- Q Capture
- TGT1
- TGT2
- TGT3
Performance of Q-Replication vs. SQL Replication - z/OS

- Q Rep
- SQL Rep

Latency (Seconds)

Workload Throughput (Rows / Second)
Subscription Types

- **Unidirectional**
  - Changes are replicated in one direction between two servers (i.e. from source to target)
  - Changes can be filtered and transformed

- **Bidirectional**
  - Changes are replicated in two directions between two servers
  - Utilizes **VALUE** based conflict detection

- **Peer to peer**
  - Changes are replicated between 2 or more servers
  - Utilizes **VERSION** based conflict detection
Q Replication Multidirectional Configurations

- **Peer-to-peer**
  - No master copy
  - Guaranteed Convergence
  - **Version** based conflict resolution
  - Requires extra columns and triggers to provide versioning of rows
  - N nodes: N * (N-1) subscriptions

- **Bi-directional**
  - One node prevails in case of conflicts
  - **Value** based conflict resolution
  - Uses old and new value data comparisons
  - 2 nodes only
Q Replication – Defining Subsets or Filters

- Subset data
  - Subset of rows through Q Capture predicate on subscription/publication
  - Subset of columns through subscription/publication definition
  - Option included for ignoring deletes
  - Signal defined to allow user selected transactions to be ignored

- Predicate examples
  - Based on values in the row data itself
    - WHERE :LOCATION = 'EAST' AND :SALES > 100000
  - Based on values in other data
    - WHERE :LOCATION = 'EAST' AND :SALES > (SELECT SUM(expense) FROM STORES WHERE stores.deptno = :DEPTNO)
Q Replication - Transformations

- Transformations achieved through:
  - Triggers on the target table
  - Publish event to User Application
  - Stored Procedures called by Apply at the row level

Apply calls Stored Procedure, mapping columns to input parms

Update Target table X where trg1 = “a”;

Stored Procedure performs logic and makes insert/update/delete
Apply Load Options

- A subscription is defined as either: automatic load, manual load, no load required

- Automatic load:
  - Load is performed by Apply, with automatic coordination of the simultaneous capture of changes, loading of the new table, and apply of changes to other tables.

- Manual load:
  - Load is performed by user, coordination is required, and will be handled by user (with some help from our administration).

- No load:
  - No loading required, no coordination required, can immediately capture and apply changes
  - Example: target system is built through backup/restore, with replication started from an inactive source
Replication Administration

- Replication Center GUI
  - Launchpads, Wizards, Online Help
  - Definitions, Operations, Monitoring

- Command Line Interface
  - Scripts or interactive mode
  - Example:

```
C:\asnclp
REPL > CREATE QSUB USING REPLQMAP ...
REPL > CREATE SUBSCRIPTION SET SETNAME ...
REPL > CREATE MEMBER IN SETNAME ...
```

- Java API’s
  - Typically used when replication is embedded
Create large numbers of subscriptions at a time!!
Table Reconciliation Utilities

- **ASNTDIFF**
  - Utility that compares a subscription’s source table (S) with its target table (T)
    - Generates a table of differences between the two
      - Rows in S but not in T
      - Rows in T but not in S
      - Rows in T and S, but with different values
    - Checksum used to compare contents of entire row
    - Very similar concept to file compares such as UNIX diff command
    - Differences can be used to change source, target, or both

- **ASNTREP**
  - Utility that uses the table built by the tdiff utility and issues SQL to make table (T) match table (S)
New in 2005 – MQ Client Support

- Distributed platforms only
- Allows separation of Database servers and MQ servers
- Allows replication support on platforms which currently lack MQ Server support
New in 2005 – Federated Targets

- Uses WebSphere II Federation
- Provides high speed parallel apply of data
- New targets:
  - Oracle, Sybase (fp 9)
  - MS SQL Server, and Informix(fp10)
Why Publish Data?

- **Application to Application Messaging**
  - Drive downstream applications or APIs based on the transactional changed data of database events
  - Reduce application development and maintenance, performance impact to source applications, and availability impact to source applications

- **Meet Auditing Requirements**
  - Capture and store information regarding what changes were made to critical business data and by whom

- **Event Notification**
  - Stream changed data information to Web interfaces
  - Stream only particular events of interest (filter data)

- **Warehouse / Business Intelligence**
  - Integrate captured changed data with an ETL tool
  - Perform very complex transformations
  - Use a specific transaction format to update target
Q Replication – Event Publication Process

**SOURCE**
- SOURCE1
- SOURCE2

**DATA SOURCES**
- DB2 Log
- Q Capture

**ADMINISTRATION**
- Replication Center
- Replication Monitor

**TARGET**
- User Application
- WBI Event Broker
- User Application
- DB2 MQ Listener
- User Stored Procedure

**SYSTEMS**
- DB2 MQ
- Listener
Event Publishing - Publication Options

- **Format**
  - Only data from committed transactions is published
  - Data is UTF-8, self describing with XML tags
  - Row based = one row per message
  - Transaction based = one transaction per message

- **Row Content**
  - Subset by column
  - Subset by predicate
  - Changed column values only or all column values
  - New data values only or include old values
  - Row sent on any change or only on published column changes
  - Suppress deletes
Information Integrator Event Publishing for Classic Sources

- Capture data changes for classic sources using log data where available
- Correlate by transactions within a single database
- Publish onto message queue in XML format

Extending the value proposition of the MQ based replication and publishing architecture
Replication and Event Publishing Products: z/OS

- **DB2 DataPropagator for z/OS**
  - DB2 UDB sources and targets (DB2 for z/OS V7 and V8)
  - SQL Replication only

- **WebSphere Information Integrator Replication for z/OS**
  - DB2 UDB sources and targets (DB2 for z/OS V7 and V8)
  - Includes SQL Replication, Q Replication, and DB2 Event Publisher

- **WebSphere Information Integrator Event Publisher for z/OS**
  - Event publishing to message queues
  - Available for DB2, IMS, or VSAM

New!!!!

WS II V8.2
Replication and Event Publishing Products: Distributed Platforms

- **DB2 LUW**
  - DB2 LUW and Informix IDS sources and targets
  - SQL Replication only

- **WebSphere Information Integrator Replication Edition**
  - Includes SQL Replication, Q Replication, and DB2 Event Publisher
  - DB2 LUW sources and targets (Q Replication) – note that Websphere MQ is bundled with this product
  - Multi-vendor sources and targets (SQL Replication)

- **WebSphere Information Integrator Event Publisher Edition**
  - DB2 LUW sources – note that Websphere MQ is bundled with this product
Combining SQL and Q Replication with Event Publishing

SQL Replication and Q Replication can co-exist
Managed at source by using multiple capture schemas
One Q Capture can handle both Publications and Subscriptions
Continuous Availability using Q Replication

- Q Replication provides a solution for continuous availability where the active secondary system is also available for other applications.
Why Use Q Replication for Continuous Availability?

- Advantages
  - Allows the fastest switchover with transactionally consistent data
  - Excellent solution for scheduled outage
    - Allows flexibility of OS level, DB level, application level, data format
    - Can be easily tested and monitored
  - Allows for database read or write activity on secondary
    - Secondary site may be used for other applications
    - Is the only solution for geographically dispersed updateable databases
  - Can supplement other HA solutions
  - Allows for lower cost hardware or platform
  - Low impact on source applications

- Disadvantages
  - Asynchronous
    - Some data is left behind in a failure scenario
  - Application awareness is required (triggers, generated always columns)
IBM Software Group

High Availability Disaster Recovery for DB2 LUW

- Log data is copied synchronously or asynchronously
- Copied data is continuously applied using forward recovery

- Production Database
  - Offers a complete solution for high availability — easy to implement, replicates the complete database
  - Will not initially support reads at secondary, partitioned tables

- Standby Database
High Availability - Q Replication compared with HADR

- **HADR**
  - Sync, async, near-sync
  - Whole DB2 database
  - DDL, DML
  - Very simple to set up and manage
  - Similar configurations only
  - No support for unlogged LOBs
  - 1 read/write site only **
  - No DPF

- **DB2 II Q based Replication**
  - Near real time async
  - Selected tables/columns
  - DML only **
  - More complex to set up and manage
  - Sites can be very different
  - Can support unlogged LOBs
  - Multiple read and/or update sites
  - DPF ok

**Current restriction only**
Replication processes and subscriptions are defined in both directions and data changes flow in both directions.

- Recursion is stopped by Capture, which reads special logged events created by Apply.
- Conflict detection is typically necessary, unless the application is carefully designed to completely avoid conflicts.
Peer to Peer Q Replication – Best Practices

- Workload balancing
  - Provides best results with high ratio of reads to writes

- Conflicts
  - Plan carefully – avoidance is the best policy
  - May occur with failovers and switchbacks
  - Consider the application impact: for database convergence, single row updates are backed out, not whole transactions

- Exceptions table
  - Understand the exceptions table – all conflicts are logged there
  - Consider a global view or replicated consolidation of exceptions tables
  - Consider a trigger on the exceptions table for additional actions that need to be performed

- Application considerations
  - Make a plan to handle serialized objects such as sequences and identity columns
  - Consider impacts to triggers and triggered actions
In many online environments OLTP data is kept separately from query/history data for better performance of both update and query applications.

This user has just made an online trade – he will keep hitting enter until he sees that the trade is complete, in this case meaning it has been replicated to the trade history database.
As new orders are entered into the order entry system, the pertinent data is captured and published into a queue. The Websphere Business Integrator Event Broker processes the queued data. A billing transaction is created and queued in one system and a shipping transaction is created and queued in another system.
- When a change to customer profile information occurs in one system, the pertinent data is captured and published into a queue.
- The Websphere Business Integrator Event Broker processes the queued data.
- Transactions are created to update the customer profile information in all other database systems, as applicable.
Other Important Sources of Information/Education

- WebSphere Information Integrator sites on the web:

- Developer Works:
  - Tutorials, whitepapers, samples available now

- IBM Education for Q Replication:
  - DW240: 3 day course without MQ basics
  - DW241: 4 day course with MQ basics included

- Redbook recently published for Q Replication, EP later this year

- Consider IBM Services as part of your implementation plan