Replicating Rdb Databases with the JCC LogMiner Loader

What we've learned along the way

JCC Consulting, Inc.

Abstract

At JCC we have developed a product that is in wide use in mission critical architectures. We have reported before on the myriad ways that it is used.

In this session, we will discuss

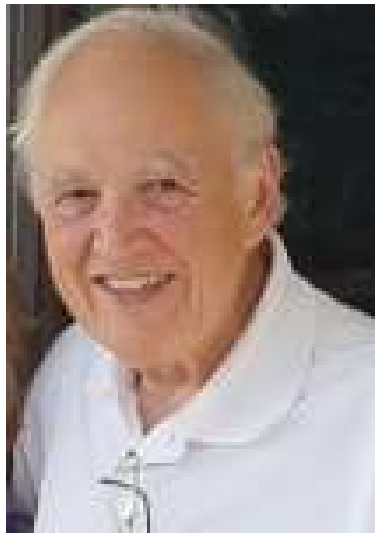
What We Have Learned Along the Way

It is our hope that you will find some insights to benefit your architectures and we hope that you will, at a minimum, find the points interesting.

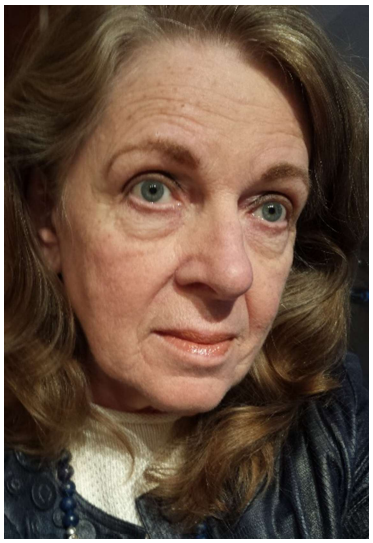
Who are we?

- JCC Consulting, Inc., began in June, 1984
- JCC first deployed the JCC LogMiner Loader in 2001
- The Loader team all do architecture, training, and ….

Tom Musson
Developer



Jeff Jalbert
Testing



Cheryl Jalbert
Documentation

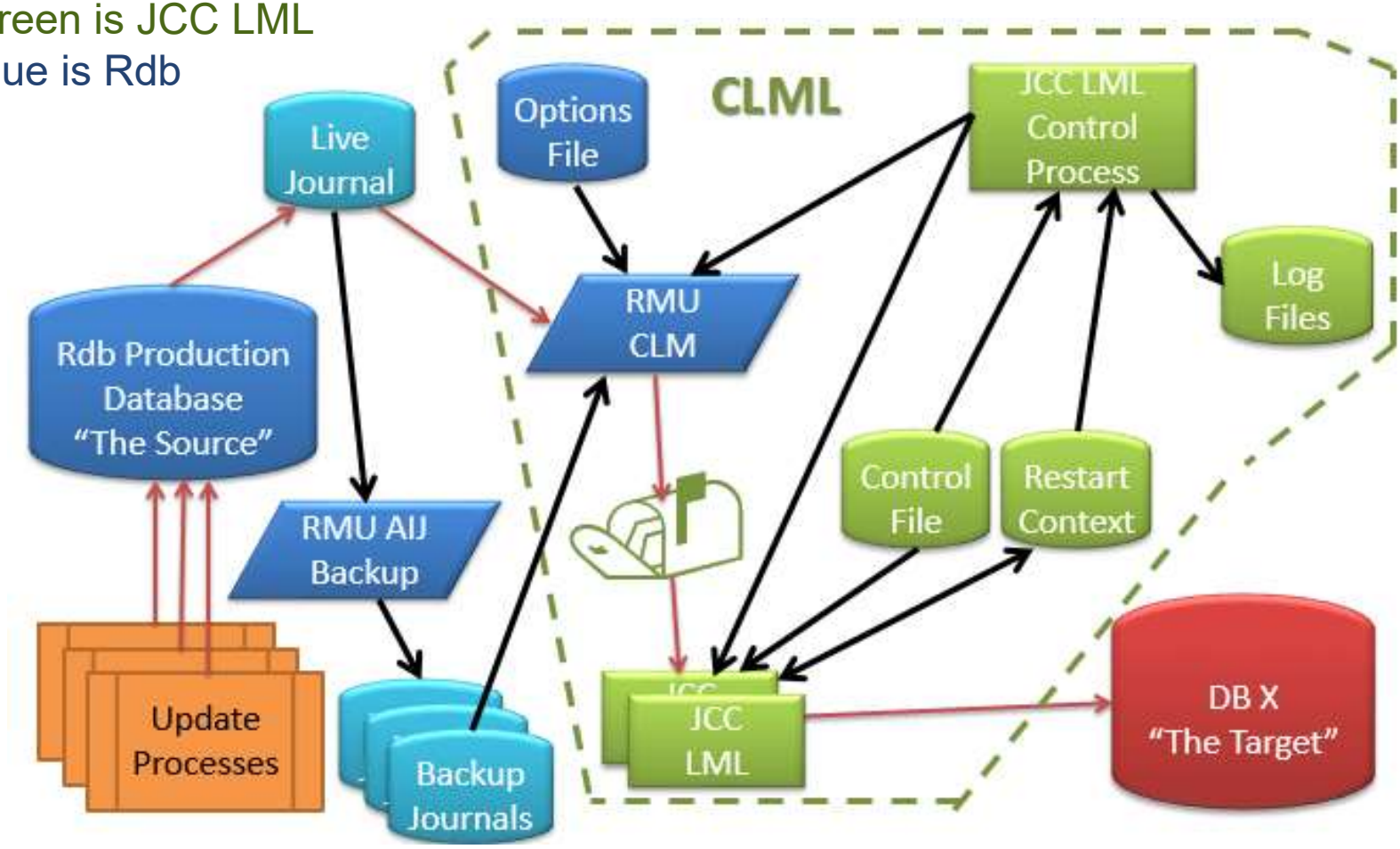
Keith Hare
Installation

Source of Obscure New Knowledge

- Some of what we've learned is because our product is middleware.
- Some of what we've learned points up the differences in database products.
- Some of what we've learned concerns differing application practices.
- Some of what we learned is because our customers are clever and think up unanticipated ways to deploy the product.

Middleware

Green is JCC LML
Blue is Rdb



Some of what we've learned…

- About the source
- Tuning replications starting on the source side
- Locking
- Tuning all the pieces, including JAVA
- AIJ Backup
- Tuning human issues
- Designing & tuning the target for performance
- Network latency
- Detailed differences among targets
- Initial load of the target
- Operational dependencies & changing metadata
- Long running or large transactions

The Rdb Source and the LogMiner

- Rdb LogMiner has a low impact on the source Rdb database.
  - Generally, those who install it say the impact is unnoticeable.

LogMiner and Locking

- The exceptions that we see involve locking interactions.
- LogMiner maintains a lock on the last location that it read in an AIJ.
- In certain circumstances, this can create a problem.
  - If a database becomes quiescent while running LogMiner, the LogMiner locking can block AIJ backup.
  - JCC's Loader has an option called Heartbeat that addresses this.
    - Heartbeat requires a table of one row.
    - Heartbeat updates that row once per interval.
    - The Loader Administrator can set the interval.
    - The update causes the LogMiner to release locks which permits Backup to acquire the locks that it needs.

Source System Support for Replication

- Changes on the source system can improve overall performance of the replication.
- The OpenVMS account used to run the Loader must have the resources that it needs.
  - Page file quota
  - Working set quota
  - Lock quota
- Enable 64-bit memory address space:

```
$ Define JCC_COMC_VA_MEMORY_MODEL 1
```

  - To take full advantage of 64-bit architecture
  - For large transactions

Locking Between Loader Threads

- Loader uses the VMS lock manager to coordinate multiple threads.
  - Each Loader family has a specific lock tree.
  - Precedes any locking that might be done on the target.
  - A lock is created for each row to be written to the target.
    - The Loader locking model is adjusted for performance.
  - Effectively is row level locking before data transmitted.
  - Avoids buried updates.
- Row level locking is not appropriate for all situations.
  - Large transactions may require large numbers of locks
    - Can adjust the granularity level to row, page, page interval or logical area
  - Some workloads may cause locking between threads or objects other than rows.
- The Administrator may change the locking level by setting a logical name, JCC_LML_LOCKING_LEVEL.

Java Tuning

For JDBC targets, JCC LML instantiates a Java Virtual Machine (JVM).

- Use logical name JCC_LML_JAVA_COMMAND_LINE to configure JVM stack and memory.
- Example:

```
$ define JCC_LML_JAVA_COMMAND_LINE "-Xmx256m -Xss1m"
```

  - "-Xmx256m" – JVM memory set to 256 megabytes
  - "-Xss1m" – JVM stack set to 1 megabyte
  - Values may need to be increased for larger transactions
- Insufficient memory and stack results in obscure exceptions
- JDBC driver specific directives can be included directly in the URL specified for the target.
  - Format is unique to each JDBC driver

AIJ Backup

- Since the LogMiner is dependent on finding and processing the AIJs, backup practices get our attention.
- AIJ files must remain available until processed.
- Heartbeat may be required to avoid locking contention between AIJ backup and LogMiner.
- Backup strategy should be carefully implemented.

Rdb Automatic Backup Server (ABS)

We recommend against using ABS.

- With ABS, backups occur at unpredictable times.
- We prefer to manage AIJ backups with command procedures scheduled at deliberate times.
- It is extremely important to control when AIJ backup occurs relative to other activities and critical times.
- When running LogMiner, it becomes even more important to avoid surprise backups.
  - Beginning the backup at essentially the same time as the LogMiner can lead to a rare – and undetected – issue.
  - When the LogMiner is processing a backup journal and backup removes the next AIJ from the live journal, processing can be interrupted with an exception message. Operations can be resumed, but require intervention.
- Blog entry http://www.jcc.com/lml-abs-bad

Suffer for Stability

That brings to mind something else we've learned along the way:

- Some companies develop rules that defeat re-thinking prior decisions.
- Some companies have politics that defeat change.
- Understanding the difficulties caused by ABS is a frequent example of Suffering being a victim of Stability.

- We've seen enterprises fail to upgrade when a new version solves an issue that is specific to the enterprise
  - JCC LogMiner Loader
  - Rdb
  - VMS
- Stability is good, rigor mortis is not.

Performance on the Target

- Rdb is a powerful engine.
- Rdb applications are often finely tuned.
- Rdb applications can produce large volumes of updates very quickly.
- LogMiner and Loader are relatively fast.
- Therefore, a target may be challenged to keep up.

Story – Slowing down the loader

- The Loader actually includes a setting that can be used in testing to modify how rapidly it attempts to update the target.
  - The capability was introduced when what was designed as a replacement for the Rdb application couldn't keep up. Developers requested that we slow down what seemed a fire hose.
  - The capability can also speed up the Loader output. This can be used to test scalability before data volumes rise beyond what the target can receive.
- Hopefully, the testing can be used to tune the target before too much is invested.

Target Primary Keys

- A primary key must be defined for the Loader to modify or delete a row on the target.
  - Replication consists of inserts, updates, and deletes.
  - Not everyone has the same interpretation of a primary key.
- A primary key is a column or columns that uniquely identify a specific row.
  - No column that is part of a primary key can change.
  - No column may be NULL.
- Timestamps can be problematic in a primary key.
  - Precision issues between source and target
- Need index on Primary Key column(s)
  - Some implementations automatically create an index on a declared primary key.
  - Some implementations require DBA to create the index.

Tuning Performance on the Target

- In most cases, an index is needed for each primary key that will be used by the Loader to find a row to delete or modify.
  - Replicating to an initially empty table can result in lock conflict, if using multiple threads.
- Additional indices on the table can slow processing
  - Or even create lock contention.
- Triggers on target table can slow processing.
- The target must be capable of absorbing the transactions to be written to it.
- The Loader Administrator or someone else will have to understand – or learn – how to tune the target chosen.

Other Design Issues for Target

- Constraints
  - The order of records in the LogMiner output cannot be predicted.
  - The Loader applies updates within a transaction out-of-order.
  - A constraint for referential integrity, for example, can cause the Loader to fail if a child row is received for input before the parent row.
  - Constraints must be carefully considered, if used at all.
- Triggers
  - A trigger that has fired in the source database does not need to be repeated in the target and can lead to erroneous data.
  - Triggers that have not fired in the source database may be needed in the target to satisfy an application need.

Target Transaction Consistency Model

- For Rdb targets, the locking model used by the Loader is read committed.
  - This is less conservative than Rdb's default serializable.
  - It reduces the duration that read locks are held.
  - It is appropriate for most circumstances.
  - It may require a retry in rare cases, but the retry is handled the same way that deadlocks are handled – rollback and retry.
    - May happen with very high update tables with particularly high duplicate cardinality indices.
- Future release allows isolation level to be controlled
  - Logical name JCC_LML_ISOLATION_LEVEL

Character Sets And Collations

Replicated data can be affected by collations on the target

- Collation encompasses several characteristics
  - Character representation – one or more bytes per character
  - Collating sequence – sort order, value comparisons
- Some ASCII characters translate to multi-octet characters
  - If the target column is defined in terms of bytes, source data may not fit
  - If the target column is defined in terms of characters, data will fit
- SQL Server default collation is case-insensitive
  - If the primary key is a character string, this can lead to performance problems

SQL Server JDBC and Unicode

- SQL Server JDBC driver, by default, converts character strings to Unicode
  - Performance issues if target primary key is character string and is not Unicode
- Use following JDBC URL to send non-unicode characters.

```
output~jdbc~synch~jdbc:sqlserver://<target node>
  ;InstanceName=<instance-name>
  ;DatabaseName=<database-name>
  ;sendStringParametersAsUnicode=false
```

  - Example of sending parameters to the JDBC driver
  - Punctuation is driver specific

Other targets and JDBC drivers may have similar quirks.

Insert Priority

- By default, JCC LogMiner Loader
  - Attempts to update target row
  - If no row found, Insert target row
- For certain Oracle workloads, more efficient to:
  - Insert target row
  - If duplicate key exception, update target row
- Enable insert first with logical jcc_lml_optimize_insert

```
$ define jcc_lml_optimize_insert 1
```

- Target must enforce unique primary key with constraint and/or indexes.

Network Latency

- LML updates of the target database are synchronous – Send a buffer and wait for a response.
- Records per second are a function of latency and buffer size.
- Rdb and OCI interfaces are a record at a time. For example:
  - If the round trip time to the target is 50 milliseconds, throughput is 20 rows per second.
  - Million row transaction will require almost 14 hours
- Most (not all) JDBC drivers support sending multiple rows in a buffer.
  - Specify batch size with logical name JCC_LML_JDBC_BATCH_SIZE

```
$ define jcc_lml_jdbc_batch_size 50
```

- Parallel threads in the Loader may help
  - Must be tuned
  - Cannot fix a single large transaction with parallel threads

Schema & Column Names

- Schema and column names may be specified in the map statements.
- Schema names may be required or schema names may not be used at all.
- Column names may be changed in the MapTable specification.
- Target column and table name length may shorter than Rdb
  - Rdb names can be 31 characters
  - Oracle names can be 30 characters

Target Schema Names

- Some JDBC drivers do metadata queries across all schemas, not just the schema associated with the login
  - Teradata
  - DB2
  - Others?
- Specify schema with logical name JCC_LML_JDBC_TARGET_SCHEMA
- Examples:
  - Specific schema name:

```
$ define JCC_LML_JDBC_TARGET_SCHEMA "MYSCHEMA"
```

  - Use only the schema associated with login

```
$ define JCC_LML_JDBC_TARGET_SCHEMA -
        "#JCCLML$TARGET_USERNAME#"
```

Case Sensitivity

- Some database products support case sensitive table and column names.
- The Loader Administrator may have to be aware as to whether ‘Abc’ in the target will be recognized as ‘ABC’.
  - Otherwise, the messages about not finding a particular column may seem mysterious.
- Loader supports mixed case by use of a delimiter specified by a logical name

Reserved Words

- Some database products have reserved words that cannot be used in column names.
- Oracle is one.
- The source name can be mapped to a different target name, if necessary.

Natural Language/National Language

- Oracle expresses some national language character sets with specific settings.
- The Loader Administrator may need to set an indicator to see appropriate interpretation of Japanese, Swedish, or other languages.
- JCC has found the setting NLS_LANG to AMERICAN_AMERICAN.WE8DEC to be useful in interpreting the traditional DEC_MCS character sets that may be used in the source.

```
$ define nls_lang "AMERICAN_AMERICA.WE8DEC"
```

This is almost identical to

```
$ define nls_lang "AMERICAN_AMERICA.WE8ISO8859P1"
```

If the source language is Hebrew, use

```
$ define nls_lang "AMERICAN_AMERICA.WE8ISO8859P8"
```

Oracle RAC and Load Balancing

- If the target DB is Oracle RAC instance
  - And load balancing is enabled
  - And multiple LML threads configured
  - Then LML threads may talk to different instances
- It is possible that:
  - Thread 1 inserts a row into Instance 0
  - Thread 2 updates the same row in Instance 1
- RAC has slight delay in replicating data to other node
  - Update against row in Instance 1 could fail because row from Instance 0 not yet replicated
- Solution – do not use load balancing for the Loader threads to an Oracle RAC target.

Lock Contention on Target

Deadlocks are possible on many target databases.

- Between Loader thread and application on target
- Between multiple loader threads
- When JCC LogMiner Loader receives a Deadlock exception
  - Rolls back the transaction
  - Retries the transaction
  - If it receives an exception N times without intervening successful transaction, exits with exception
    - N is determined by <retry attempts> in OUTPUT_FAILURE configuration directive
    - OUTPUT_FAILURE~<timeout seconds>~<retry attempts>

Initial Load

- The JCC LogMiner Loader publishes to the target updates made to the source.
- The target must be defined and populated with the initial set of data.
- The Data Pump that is packaged with the Loader is the favorite tool for the initial load.
- Performance of the load can be tuned.

Improving Performance for Initial Load

- When initially populating data via the Loader and Data Pump, the source database is often quiescent
  - Configure the MapTable statements to do AUDIT
  - Do not include a primary key in the MapTable definitions
  - Do not index the table
- This is also discussed in a blog post.
  http://www.jcc.com/lml-initial-load

Query Strategies & the Initial Load

- If the target tables start empty, query optimizer may choose sequential strategy.
- After several thousand rows, sequential is no longer optimal.
  - Collect Optimizer statistics.
    - Rdb – RMU/Collect Optimizer_statistics
    - Oracle –DBMS_STATS package
    - MySQL – innodb_stats_method
    - SQL Server – Update Statistics
    - Similar capabilities with other databases
  - Shutdown loader session.
  - Restart loader session.

Story – Operational Dependencies

- A client “recreated” several tables on a regular basis.
- The client restarted LogMiner and the LogMiner Loader every Monday.
- Along in June, someone questioned the restart and no one remembered why. They eliminated the “unnecessary” restart.
- In September, someone noticed some tables weren’t being replicated.
- Why?

Physical ID for the Table

- The LogMiner and the Loader work with a physical view of the database, not the logical view that we think of for relational databases.
- Both tools must provide the reference to the physical database that we are not accustomed to needing.
  - In this case, the table is identified by the LogMiner with “relation id”.
  - The Loader must understand the version of the row that it receives.
  - The Loader kit includes tools to help create the LogMiner options file and the Loader Control File without having to know all the detail.
- When the tables were dropped and recreated, the table name stayed the same but the relation id changed and the LogMiner no longer identified the table as one to process.
- People who have accepted that relational databases can be referenced by table names and such find it difficult to recognize a different approach.

Changing Metadata

- More frequently, the issue is a change in a table definition.
- If a metadata change does not touch – directly or indirectly – any column or table that is represented in the LogMiner options file or in the Loader Control File, there is no issue.
- If the metadata change does change something defined for the LogMiner Options File or the Loader Control File, it is necessary to be aware of the internal definitions so that they may be properly updated.

Steps to Change the Metadata

- Shut down the application.
- Process all existing AIJs.
- Backup AIJs.
- Shut down the Loader.

Stop Phase

- Make the metadata changes.
- Update the Control File and/or Options File.
- If necessary, update the target metadata.
- Store the backed up AIJs such that they will not be found by JCC_AIJ_BACKUP_SPEC (because they include the old definitions).

Change Phase

- Restart CLML in the live journal.
- Restart the application.

GO Phase

Long Running/Large Transactions

- Replication cannot start until the source transaction commits.
  - Adds a delay to the Loader
- Larger transactions use more memory

Blogs

- www.jcc.com/lml-blog includes a growing number of topics.
  - Managing LogMiner performance
  - Dangerous interaction between RMU AIJ Backup & LogMiner
  - Network Latency
  - … and others

Summary

In working with middleware, a wide range of client apps, and a variety of end targets, we have learned some interesting things.

- Rdb applications can be more powerful and robust than their users recognize.
- Replicating to a target requires understanding how to tune the target such that the update stream does not become a fire hose.
- Locking strategies must be tuned to the circumstances.
- Backup strategies must be managed.
- Developers who use a wide range of products must be aware of a zany collection of details of character sets, representations, case sensitivity, and other product variances.
- Network latency cannot be ignored in planning the architecture.
- Long running or large transactions can require different approaches.

Learning More

- The product is The JCC LogMiner Loader.
  - Read about it at http://www.jcc.com/lml
  - Check out the blogs http://www.jcc.com/lml-blog
  - Ask for a temporary license
- Contact us at Info@JCC.com

Join the Conversation

Join the worldwide Rdb community. Send mail to OracleRdb-request@JCC.com.

Include “SUBSCRIBE” in the body of the message.