

# JCC LogMiner Loader Version 2.0

Jeffrey S. Jalbert, JCC Consulting, Inc.

# Abstract

JCC has announced version 2 of the JCC LogMiner Loader product. This version adds a large number of features to the Loader including the ability to process larger tables and very high update databases. This presentation offers an overview of the new features in the Loader and supplies some performance data from actual customer applications.

# Agenda

- What the Loader can be used for today
- New Features in version 2.0
- Performance data
- Futures

# Current Deployment

- Replicate selected tables or entire databases to Rdb and Oracle targets – in near real time
  - Create reporting database to exploit capabilities of Oracle
  - Create image database to allow more query support for Rdb
- Publish database events to the enterprise and so trigger actions by other software, e.g. Order Fulfillment
- Create an audit history of changes to a database

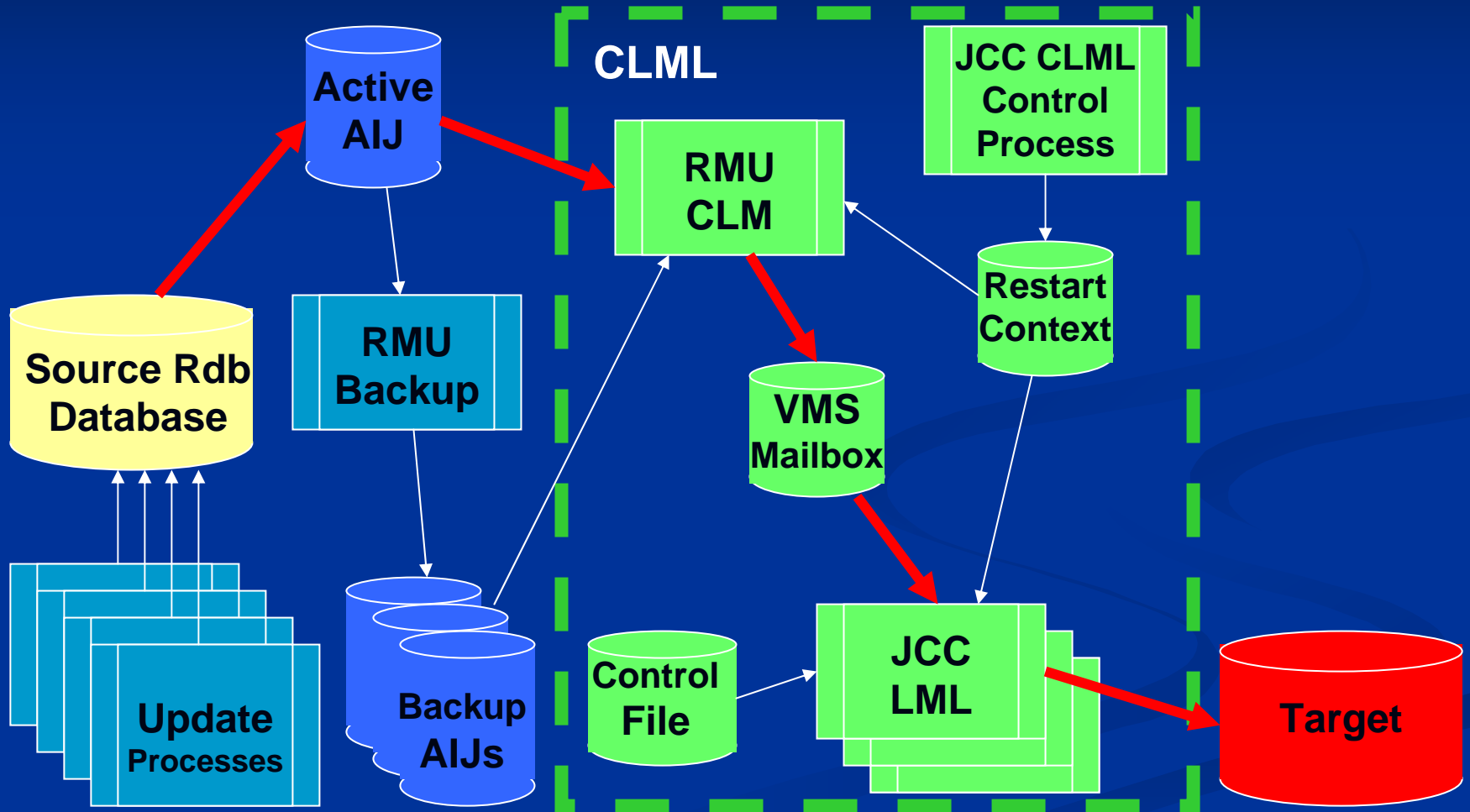
# Current Deployment

- Roll up changes from several databases to a single reporting database
- Replay changes to a database to allow recursive performance tuning
- Reorganize databases with little actual down time

# Loader Targets

- Rdb
  - Oracle
  - Your code in XML format
  - Tuxedo, new in version 2.0
- 
- Note that the format of the target may be completely different than the format of the source database

# How The Loader Does It



# Controlling the Loader Behavior

- The Loader actions are determined by a “Control File”
- Text editable format
- Using keywords you define:
  - Target
  - Behavior
  - Source database metadata
  - Most have default values

# Identifying Rows in the Target

- Changes are applied to target databases (Oracle or Rdb) on the basis of “Key” values
- Keys are data columns in target
- For databases modified by unusual mechanism (e.g. changing a column in the primary key) can use the row DB-key

# New Features in V 2.0

- The focus of 2.0 development has been on:
- Reliability
- Performance
- Increased number of columns in tables
- New target → Tuxedo
- Better monitoring
- Better DBA tools
- Complete set of examples
- Oracle 9.0 SQL\*net on VMS
- Heartbeat

# Reliability

- New exhaustive regression testing package
- Random workload
- All capabilities exercised at the same time
- Loader families are “attacked” to validate recovery and restart
- Complete differences run between source and all targets
- Cycles every 1/2 hour and runs for days (exercises the reliability of the targets)

# Performance – Multiple Threads

- Support for update intensive databases via multiple (up to 32) concurrent threads
- Thread count is dynamic to support varying workload
- DBA can force changes to thread counts
- Can be configured to avoid burying updates to target databases
- Threads can run unconstrained and materialize the Loader Sequence Number column to mark relative age of rows in the target

# Performance – Rdb Targets

- Better interface to Rdb SQL
- Old version is available if sending data to a down version of Rdb

# Performance – Commit Intervals and Checkpointing

- Multiple source transactions bundled into single target transaction
  - By default, exclude NoWork transactions from checkpoints thereby reducing the workload in the target
  - Timeouts possible if source database becomes dormant in the midst of a commit interval
  - Dynamic adjustment of checkpoint interval depending on the volume of data in transactions

# Other Performance Features

- Control placement of sortwork files
- Mechanism to define logical names within context of different processes
- On restart, detects whether the last checkpoint is in live AIJs and restarts there rather than in backups
- Additional materialized columns to facilitate physical organization of target database (Random value, Modulo, Thread number)

# Tuxedo Target

- Requires Tuxedo workstation client software on VMS server (no license fee, except client licenses)
- Can configure message size to control network load
- Dynamic fail-over if target fails
- Dynamic load balancing if Tuxedo application hosted on several servers
- Works within Tuxedo Transactions
- TP calls or enqueues as specified in the loader control file

# Monitoring and DBA Tools

- Display the current Loader locking tree for all families or a single family
- Display current checkpoint information
  - Place in DCL symbols
  - Useful for restart when combined with RMU
  - Or if you have some sort of disaster

# Monitoring and DBA Tools

- Enhanced Logging by Loader threads
  - Can provide periodic updates of work accomplished
- Control of Loader families cluster wide
  - Requires DECnet proxies
- Enhanced monitoring
  - Full Report now displays costs for each stage of operation
  - Two new displays
    - Detailed – 24 line display of what is going on
    - Comma Separated Values [CSV] suitable for loading into a database or spreadsheet

# Heartbeat

- Some databases are infrequently updated
- LogMiner then stalls AIJ backup process
  - Can time out AIJ backups and lead to operational overhead
  - DBA may introduce some artificial update mechanism
- Solution is for the CLML process to write to a private heartbeat table at timed intervals
  - Optional feature not required for all databases
  - Function is disabled if any problem detected making it work
    - OPCOM message sent signaling this event

# Performance Statistics Example

- Loader being used to roll up 28 source databases to a single target [Tuxedo]
- Most of time spent in waiting for Tuxedo servers to respond
  - Target servers are a pair of Himalaya systems
- Maximum throughput in excess of 2,400 rows per second
  - Keeps up with 40 Wildfire 1Ghz CPUs running application

# More Performance Info

- Target Tuxedo servers were down for 3 days
- Catch-up of one database
  - 11 tables being LogMined
  - 173 backup AIJs
  - 13,019,675 blocks of AIJ
  - 644,499 transactions 15,939 were NoWork
  - 2,970,068 rows sent to the target
  - 1,144,352 rows were “filtered” by the Loader

# Catch-up Performance

- Started at 10:25
- Ended at 12:04 – 1 hour and 39 minutes later
- At peak was sending almost 2,000 rows per second
- LogMiner used 30% of 1 CPU
- Loader threads each used 4% of 1 CPU for a total of 120%
- System still had idle time



# Futures

- Data Pump, forces rows into the AIJ without making changes
- SQL-like predicate evaluation for filters
- Support no-change behavior to avoid an update in target
- Materialized columns as part of the target table key
- Transaction type selection for Rdb targets
- Real-time throttle
- Materialize table for commit records
- Performance enhancements
- Oracle 9.2 SQL\*net of VMS
- Send single row to multiple target tables
- More database targets

# Availability

- Kit is available at [FTP.JCC.COM](http://FTP.JCC.COM)
  - Documentation
  - Kit
- Evaluation license available on request
  - Send mail to [jeff@jcc.com](mailto:jeff@jcc.com)

# Acknowledgements

- Thanks to Rdb engineering for their support and counsel

# Questions

