Replicating Rdb Databases with the JCC LogMiner Loader

Keith Hare, Tom Musson, Jeff Jalbert, Cheryl Jalbert
JCC Consulting, Inc.

Abstract

The JCC LogMiner Loader supports replicating committed transactions from Rdb databases to other databases using a variety of transports including Rdb, Oracle OCI, and JDBC drivers.

This session provides a detailed example of setting up replication to a Postgres database on a Linux server. It also describes new JCC LogMiner Loader features and how to use them.

Introduction

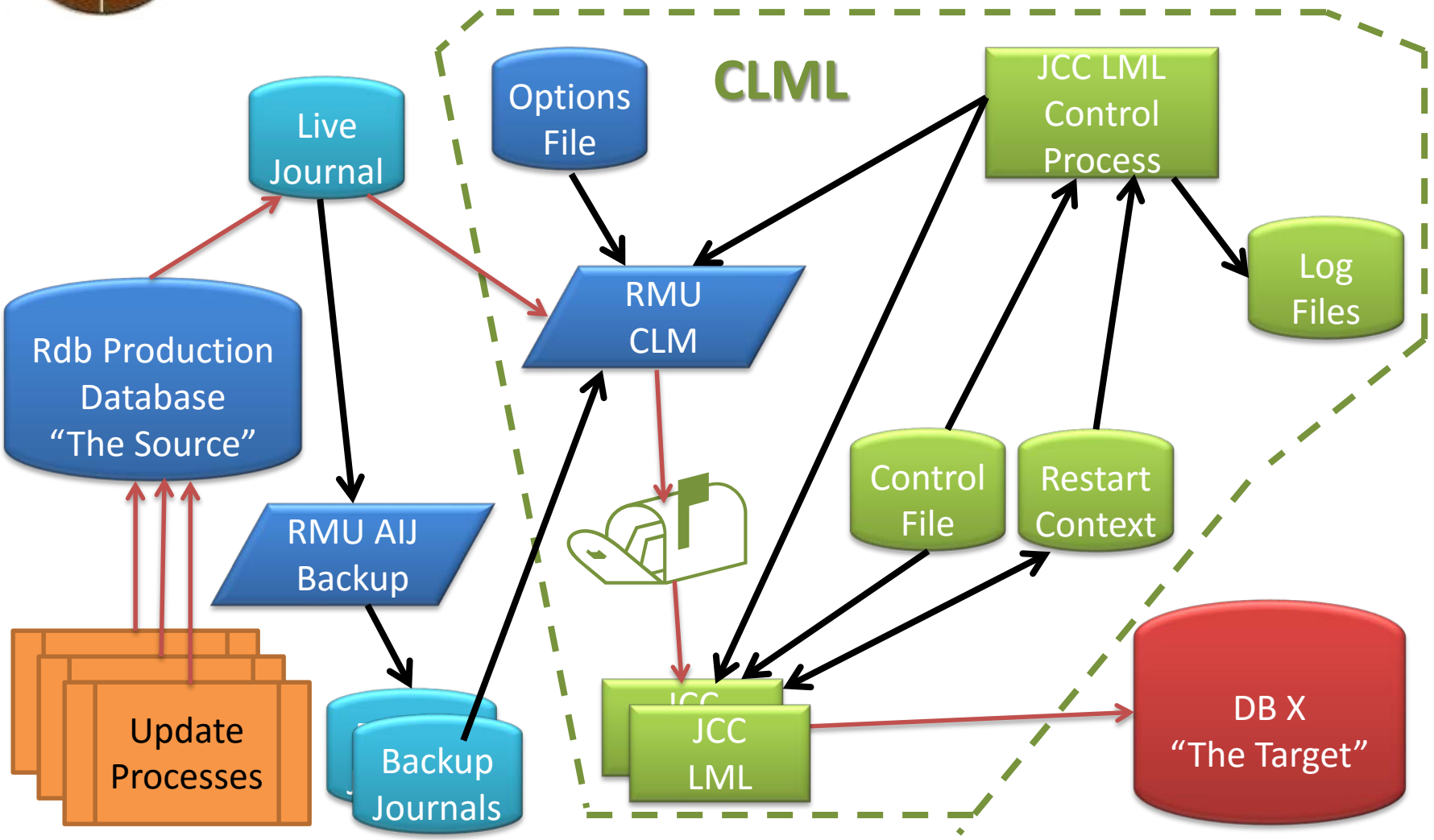
- Data Replication Goals
- The Big Picture
- Preparing the source database
- Preparing the target database
- Setting up the JCC LogMiner Loader
- Monitoring Replication
- But what about …?
- Summary

Data Replication Goals

Make Rdb data available another environment

- Near “real time”
- Transactionally consistent
- Reporting, Ad-Hoc queries, web site, testing, Audit, etc.

The Big Picture



Sources and Targets

- Source
  - Oracle Rdb (any version that supports the LogMiner)
- Target
  - Oracle Rdb (any version that supports multi-statement procedures)
  - Oracle (requires SQL*net client on the system running the Loader)
    - 12.1
    - 11.0, 11.1
    - 10.1, 10.2
    - 9.0, 9.2
    - 8.1 - deprecated
    - Oracle client-server compatibility (see Oracle matrix)
  - JDBC Class 4 driver
  - Tuxedo
  - XML (to your own API)
  - File

Sources and Targets (cont.)

- Format of the target can be completely different from format of the source database.
  - Logically
  - Physically
- Tuning of the target can be different.
  - Indices
  - Placement
  - Buffering
  - Caching

JCC LogMiner Loader Example

- Source Database
  - Oracle Rdb V7.3-121
  - OpenVMS Alpha V8.4
  - Virtual Alpha ES40, 1 CPU, 1,024 MB memory
- Target Database
  - Postgres V9.4.4
  - Centos V7.1 (Linux)
  - Virtual X86, 2 CPUs, 6,144 MB memory

Preparing the source database

- Oracle Rdb V7.0.6.5 or later
  - OpenVMS V7.1 or later on Alpha or Itanium
  - More recent versions of Rdb and OpenVMS support more options
- Circular AIJs enabled
- Continuous Logminer enabled
- Immutable primary keys

Create Journal Example

```
$ define mfp_db JCC_ROOT:[KEITH.SQL_CLASS.mf_v73]mf_personnel
$ rmu/set after /reserve=10 mfp_db
$ rmu/set after mfp_db                    -
/backups=edit_filename=("_",sequence)     -
/allocation=20000     -
/add=(name=aij_01       -
     ,file=jcc_root:[KEITH.SQL_CLASS.mf_v73.aij]MFP_AIJ_01 -
     ,allocation=20000                                     -
    )                                                      -
...
/add=(name=aij_04                                          -
     ,file=jcc_root:[KEITH.SQL_CLASS.mf_v73.aij]MFP_AIJ_04 -
     ,allocation=20000                                     -
    )
$ rmu/set logminer/enable/continuous mfp_db
$ rmu/set after/enable mfp_db
```

Immutable primary keys

- Does not have to be defined as the source primary key
- Must not be updated
- Used for the target table primary key
- Can add SQL identity column to source tables if nothing else available

Target

- Creating tables
  - Primary Key and indexes
- Querying Data
- Describe a table
- This example uses Postgres on Centos 7 Linux

Preparing the target database

- Command line SQL Utility
  - Invoke command line SQL with “psql”
  - Exit with “\q”
- Example

```
[keith@centos7-kwh ~]$ psql
psql (9.4.4)
Type "help" for help.

keith=# \q
[keith@centos7-kwh ~]$
```

- Postgres SQL looks a lot like Rdb SQL

Create Table Example

- Script created with RMU/EXTRACT/ITEM=All

```
keith=# create table WORK_STATUS (
keith(#     STATUS_CODE
keith(#         CHAR (1),
keith(#     STATUS_NAME
keith(#         CHAR (8),
keith(#     STATUS_TYPE
keith(#         CHAR (14))
keith-#     -- comment on table WORK_STATUS is
keith-#     -- information related to work status codes
keith-# ;
CREATE TABLE
keith=# alter table WORK_STATUS
keith-#   add constraint Work_Status_PK
keith-#       primary key (status_code)
keith-# ;
ALTER TABLE
```

Querying Data

- SQL is SQL

```
keith=# select last_name, first_name, employee_id, birthday
keith-# from employees
keith-# where employee_id <= '00170';
   last_name    | first_name | employee_id |  birthday
----------------+------------+-------------+------------
 Dietrich       | Rick       | 00166       | 1954-03-20
 Kilpatrick     | Janet      | 00167       | 1937-03-05
 Nash           | Norman     | 00168       | 1932-10-23
 Wood           | Brian      | 00170       | 1957-06-03
 Toliver        | Alvin      | 00164       | 1947-03-28
 Gray           | Susan      | 00169       | 1938-08-13
 Smith          | Terry      | 00165       | 1954-05-15
(7 rows)

keith=#
```

Describe a Table – \d

```
keith=#  \d employees
          Table "public.employees"
     Column     |     Type      | Modifiers
----------------+---------------+-----------
 employee_id    | character(5)  | not null
 last_name      | character(14) |
 first_name     | character(10) |
 middle_initial | character(1)  |
 address_data_1 | character(25) |
 address_data_2 | character(20) |
 city           | character(20) |
 state          | character(2)  |
 postal_code    | character(5)  |
 sex            | character(1)  |
 birthday       | date          |
 status_code    | character(1)  |
Indexes:
    "employees_pk" PRIMARY KEY, btree (employee_id)
    "employees_first_name" btree (first_name, employee_id)
    "employees_last_name" btree (last_name, employee_id)
```

Setting up the JCC LogMiner Loader

- JDBC Driver
- Option File
- Configuration File
- Command procedure

FTP JDBC Driver to OpenVMS

- Volume is ODS-5 so supports lowercase names

```
dba08 > di
Directory JCC_ROOT:[KEITH.POSTGRES]
postgresql-9_3-1103_jdbc3.jar;1     1056/1062     15-SEP-2015 21:49:32.81
postgresql-9_4-1201_jdbc4.jar;1     1256/1260     15-SEP-2015 21:49:32.11
postgresql-9_4-1201_jdbc41.jar;1    1267/1269     15-SEP-2015 21:49:32.23
Postgresql-JDBC-Driver-explaination.txt; 1/9      15-SEP-2015 21:49:26.08
Total of 4 files, 3581/3600 blocks.
```

- Must reset the file attributes

```
dba08 > set file/attr=(rfm:stmlf,rat:cr,lrl:0,mr:0) *.jar;*/log
%SET-I-MODIFIED, JCC_ROOT:[KEITH.POSTGRES]postgresql-9_3-1103_jdbc3.jar;1
modified
%SET-I-MODIFIED, JCC_ROOT:[KEITH.POSTGRES]postgresql-9_4-1201_jdbc4.jar;1
modified
%SET-I-MODIFIED, JCC_ROOT:[KEITH.POSTGRES]postgresql-9_4-
1201_jdbc41.jar;1 modified
dba08 >
```

Which Postgres JDBC Driver?

Depends on Java Version

- Java 1.7 or 1.8
  - postgresql-9_4-1201_jdbc41.jar
- Java 1.6
  - postgresql-9_4-1201_jdbc4.jar
- Java 1.5
  - postgresql-9_3-1103_jdbc3.jar

Sidetrack – Java on OpenVMS

- Alpha OpenVMS V8.4

```
dba08 > java -version
java version "1.5.0"
Java(TM) 2 Runtime Environment, Standard Edition
Fast VM (build 1.5.0-8, build J2SDK.v.1.5.0:02/08/2012-17:12, native
threads, jit_150)
dba08 >
```

- IPF OpenVMS V8.4-1H1

```
selene > java -version
java version "1.6.0"
Java(TM) SE Runtime Environment "1.6.0-6"
Java HotSpot(TM) 64-Bit Server VM (build 20.6-b06, mixed mode)
selene >
```

Generate Options File

Tells RMU/Unload/After/Continuous which tables to unload

```
dba08 > JCC_CREATE_LOG_MINER_OPT_FILE mfp_v73
%SQL-F-RTNNOTDEF, function or procedure JCC$SET_SYMBOL is not defined
JCC-I-VMS_FUNCTIONS, VMS_FUNCTIONS not in database.  Exclude capability
disabled.
...
```

- Exception message does not matter for this example
- Options file only needs to be regenerated when tables added

Options File Example

- Generated by the previous example

```
!
!  Continuous Logminer Options file
!  Generated at 2015-09-16 19:40:15
!
table=CANDIDATES,output=rdb_logminer_output_file
table=COLLEGES,output=rdb_logminer_output_file
table=DEGREES,output=rdb_logminer_output_file
table=DEPARTMENTS,output=rdb_logminer_output_file
table=EMPLOYEES,output=rdb_logminer_output_file
table=JOBS,output=rdb_logminer_output_file
table=JOB_HISTORY,output=rdb_logminer_output_file
!table=RESUMES,output=rdb_logminer_output_file
table=SALARY_HISTORY,output=rdb_logminer_output_file
table=WORK_STATUS,output=rdb_logminer_output_file
```

- RESUMES contains a segmented string

Generate Table and MapTable

```
dba08 > JCC_LML_CREATE_CONTROL_FILE mfp_v73 all
...
```

- Creates three files
  - Table File – MF_PERSONNEL_TABLE.INI
    - Describes source tables
    - Must be regenerated if source tables changed
  - MapTable – MF_PERSONNEL_MAPTABLE.INI
    - Maps source tables to target tables
  - DateFilter – MF_PERSONNEL_DATEFILTER.INI
    - Useful if source might have bad dates

Table File

- Describes source data

```
!
!  JCC LogMiner Loader Metadata (DDL) Control File
!  Generated at 2015-09-16 19:59:35
!
Table~CANDIDATES~1~~NoMapTable
Column~CANDIDATES~LAST_NAME~1~14~0~14~0
Column~CANDIDATES~FIRST_NAME~2~10~0~14~0
Column~CANDIDATES~MIDDLE_INITIAL~3~1~0~14~0
Column~CANDIDATES~CANDIDATE_STATUS~4~255~0~37~0
!
...
```

- Table name and version
- Columns with datatype information
- Do not generate by hand

MapTable – Candidates

```
...
MapTable~CANDIDATES~CANDIDATES~Replicate
!
!Table_"CANDIDATES"_has_neither_a_primary_key_nor_a_unique_index.
!
MapColumn~CANDIDATES~LAST_NAME
MapColumn~CANDIDATES~FIRST_NAME
MapColumn~CANDIDATES~MIDDLE_INITIAL
MapColumn~CANDIDATES~CANDIDATE_STATUS
!
!Originating_DBKey_column_required_due_to_no_key.
!
MapColumn~CANDIDATES~originating_dbkey
MapKey~CANDIDATES~originating_dbkey
...
```

- Primary Key is originating_dbkey

MapTable – Employees

```
MapTable~EMPLOYEES~EMPLOYEES~Replicate
!
!Using_defined_primary_key_constraint.
!
MapColumn~EMPLOYEES~EMPLOYEE_ID
MapColumn~EMPLOYEES~LAST_NAME
MapColumn~EMPLOYEES~FIRST_NAME
MapColumn~EMPLOYEES~MIDDLE_INITIAL
MapColumn~EMPLOYEES~ADDRESS_DATA_1
MapColumn~EMPLOYEES~ADDRESS_DATA_2
MapColumn~EMPLOYEES~CITY
MapColumn~EMPLOYEES~STATE
MapColumn~EMPLOYEES~POSTAL_CODE
MapColumn~EMPLOYEES~SEX
MapColumn~EMPLOYEES~BIRTHDAY
MapColumn~EMPLOYEES~STATUS_CODE
MapKey~EMPLOYEES~EMPLOYEE_ID
```

MapTable Comments

- Often requires editing
  - Correct primary key
  - Table or Column names different in target
- MapKey specifies logical primary key
  - Used when Updating/Inserting target
  - Generated based on:
    - Primary key constraint on source
    - Unique index
    - Otherwise add OriginatingDBKey column

MapTable – Originating DBKey

- Originating DBKey can be used as primary key for target table
- Rdb DBKey is physical address of source record
- Use 8-byte integer (BIGINT) in the target DB
- If source table is reorganized (unload/load) target table must be completely reloaded

Date Filter File

```
!
!  JCC LogMiner Loader DateFilter Control File Template
!  Generated at 2015-09-16 19:59:38
!
FilterMap~EMPLOYEES~where \
COALESCE(BIRTHDAY,date vms'17-NOV-1858') between date vms'17-NOV-1858'
and date vms'31-DEC-9999'
...
```

- OpenVMS Date datatype supports dates well past the year 9999
- Date Formatting functions support 4 digit years
- Eliminate rows with bad dates
- Optional but useful for some environments

Datatypes

- Most Rdb datatypes have corresponding Postgres Datatypes
- Some Rdb datatypes need to use different Postgres datatypes
  - For Scaled Integers (INTEGER(M)) use
    - Number(N,M)
    - Money
  - For VMS Date use
    - Timestamp(2)

Putting it together – MFP_POSTGRES.COM

- Sets up environment
  - JCC_LML_USER and JCC_LML_JDBC_USER
  - Various Logical Names
- Invokes **JCC_RUN_CLM_LML** with 3 parameters
  - **MFP_V73 – Logical** name pointing to the source DB
  - **MF_PERSONNEL_LM_UNL.OPT** – Lists tables for RMU/UNLOAD/AFTER/Continuous to extract
  - **MF_PERSONNEL_POSTGRES.INI** – main configuration file for the loader session. Includes additional files:
    - **MF_PERSONNEL_TABLE.INI** – describe source
    - **MF_PERSONNEL_MAPTABLE.INI** – maps to target db

Command Procedure – MFP_POSTGRES.COM

```
$ set verify
$ set defa JCC_ROOT:[KEITH.SQL_CLASS.LML_POSTGRES]
$ @jcc_tool_com:jcc_lml_user s
$ jcc_lml_jdbc_user 1.5.0 sys$common:[java$150.com]java$150_setup.com
fast
$ java -version
$ define jcc_lml_jdbc_batch_size 20
$ define decc$fd_locking true
$ define JCC_LML_JAVA_COMMAND_LINE " -Xmx256m -Xss1m"
$ define JCC_LogMiner_Loader_Name                KWH_POSTGRES
$ define JCC_LogMiner_Loader_HW_Response         CREATE
$ define jcc_lml_jdbc_target_schema "public"
$ define loader_postgres_jdbc_chkpt     -
     []loader_chkpt_postgres_mfp.jcclml_checkpoint
$ define JCC_LML_CASE_SENSITIVE_TARGET 1
$ define nls_lang "AMERICAN_AMERICA.WE8ISO8859P1"
$ define jcc_aij_backup_spec      -
       JCC_ROOT:[KEITH.SQL_CLASS.MF_V73.backups]*.aij;*
$ define JCC_ADD_CLM_IGNORE_OLD_VERSION_TABLES "*"
$ jcc_run_clm_lml mfp_v73 mf_personnel_lm_unl.opt     -
        mf_personnel_postgres.ini
```

Main Configuration File – MF_PERSONNEL_POSTGRESS.INI

```
logging~initialization
logging~statistics~runtime,timer,commit
logging~input~ignore_unknown_tables,log_restart
sort~by_record
input~ipc
checkpoint~1~lml_internal~asynch~loader_postgres_jdbc_chkpt
input_failure~10
output~jdbc~synch~jdbc:postgresql://192.168.27.22:5432/keith
jdbc~connect~jdbc:postgresql://192.168.27.22:5432/keith
validation~keith~<password>
jdbc~driver~org.postgresql.Driver
jdbc~classpath~/jcc_root/keith/postgres/postgresql-9_3-1103_jdbc3.jar
output_failure~1~10
parallel~1~1~constrained
thread~startup~10
thread~shutdown~1
include_file~mf_personnel_table.ini
include_file~mf_personnel_maptable.ini
```

Postgres Connect String

```
jdbc~connect~jdbc:postgresql://192.168.27.22:5432/keith
```

- jdbc:postgresql: from Postgres documentation
- Postgres installed on node 192.168.27.22
  - Centos 7 Linux running under Oracle Virtual Box
  - Could have added to Hosts file
- Postgres default port: 5432
- Database name: keith

Starting the Loader Session

- Submit the command procedure as a batch job

```
$ submit/noprint/notify/log=[] mfp_postgres.com
```

- Sets up three (or more) processes

```
dba08 > sho sys/proc=*kwh_post*
OpenVMS V8.4  on node DBA08  20-SEP-2015 08:58:07.68   Uptime  0 17:36:46
  Pid    Process Name    State Pri   I/O     CPU       Page flts  Pages
0000022F CTL KWH_POSTGRE LEF    6  67393  0 00:00:06.66    7425    724  B
00000230 ||0 KWH_POSTGRE LEF    6   7513  0 00:00:00.93    1043    458  S
00000231 CLM KWH_POSTGRE LEF    6   1539  0 00:00:01.80    2840    995  S
dba08 >
```

  - CTL process controls and logs
  - CLM process reads from the source DB
  - Thread (||0) process writes to the target DB

Loader Session Running

Rdb

```
SQL> select * from work_status;
 STATUS_CODE   STATUS_NAME   STATUS_TYPE
 0             INACTIVE      RECORD EXPIRED
 1             ACTIVE        FULL TIME
 2             ACTIVE        PART TIME
3 rows selected
SQL> update work_status
cont> set status_type = status_type;
3 rows updated
SQL> commit;
SQL>
```

Postgres

```
keith=# select * from work_status;
status_code | status_name | status_type
------------+-------------+-------------
(0 rows)
```

```
keith=# select * from work_status;
status_code | status_name |  status_type
------------+-------------+---------------
 0          | INACTIVE    | RECORD EXPIRED
 1          | ACTIVE      | FULL TIME
 2          | ACTIVE      | PART TIME
(3 rows)
keith=#
```

Log Files

- Log files are created for each of the processes
  - MFP_POSTGRES.LOG
  - JCC_TOOL_LOGS: jcc_run_clm-KWH_POSTGRES.log
  - JCC_TOOL_LOGS: jcc_run_lml-KWH_POSTGRES_0.log
- Containing logging information useful for understanding what went wrong
- Look for the first exception in a log
- JCC LML Support will usually ask for all logs

Monitoring Replication

- JCC_LML_STATISTICS Parameters
  - loader-family-name
  - [Refresh seconds]
  - [Type of logging]
    - brief|full|detail|csv|t4
  - [tardy threshold [operator class]]
- Example

```
dba08 > jcc_lml_statistics KWH_POSTGREs 6 d
```

JCC_LML_STATISTICS Detail

```
Rate:   6.00                        KWH_POSTGRES          18-SEP-2015 18:19:58.72
================================================================================
   Input: 18-SEP-2015 18:18:21.39              Output: 18-SEP-2015 18:18:21.32
--[Trail:    1.6m]--------------            ---[Trail:    1.6m]--------------
Transactions                 1424            Checkpoints                   1420
Records                     14367               Timeout                       1
   Modify                    2843               BufferLimit(    13)           1
   Delete                   10100               NoWork                        0
   Commit                    1424            Records(   1)                12940
 Discarded                                       Messages(  N/A )             N/A
   Filtered                     0               Filtered                      0
   Excluded                     0            Failure                          0
   Unknown                      0            Timeout                          0
   Restart                      4         - Current ---------------- Ave/Second -
   NoWork                       2         Checkpoints            44          7.33
   Heartbeat                    0         Records                90         14.99
Timeout                        27         Rate                43.22%
--- Restart Context ------                - Latency(sec) ------ LML detail ------
M|AIJ#                    32 |            CLM    1.6m |  Inpt  16.1%  Cnvt  27.3%
Q|VBN                  5376 |            ------------   Sort   0.0%  Trgt  54.3%
P|TSN                 74638 |            LML    0.12    Sync   0.0%  Ckpt   2.3%
 CTSN                 74638 |         - Loaders - 0 -------------------------------
  LSN                  1431 |         - States  - >
```

Statistics Description

- Input
  - 1424 Source Transactions
  - 14,367 Records
    - 2843 rows Modify (Insert, Update)
    - 10100 rows Deleted
    - 1424 Commit records
- Output
  - 1420 Checkpoints (Commits on target)
  - 12,940 Records sent to target

JCC_LML_STATISTICS – Latency

- Bottom right corner of Detail display

```
      - Current ---------------- Ave/Second -
      Checkpoints              44          7.33
      Records                  90         14.99
      Rate                  43.22%
      - Latency(sec) ------ LML detail ------
      CLM    1.6m |  Inpt  16.1%  Cnvt  27.3%
      -----------   Sort   0.0%  Trgt  54.3%
      LML    0.12   Sync   0.0%  Ckpt   2.3%
- Loaders - 0 -------------------------------
- States  - >
```

- Current is snapshot of most recent interval
- Latency tells where time is spent
  - 54.3% waiting for target is reasonable
  - If > 90%, need to tune target

But what about …?

- Shutdown and Restart
- Adding additional information
- Filtering data
- Transforming data
- Using multiple update threads
- Verifying data in target
- Additional targets

Shutdown

- jcc_clml_shutdown <loader-name>
- Example:

```
dba08 > jcc_clml_shutdown kwh_postgres
dba08 >
Job MFP_POSTGRES (queue DBA08_BATCH, entry 1) completed
dba08 >
```

Restart

- Restart information written to either:
  - High-water table (Rdb and Oracle OCI only)
  - Checkpoint file on source
- Example

```
dba08 > di *.*check*
Directory JCC_ROOT:[KEITH.SQL_CLASS.LML_POSTGRES]

LOADER_CHKPT_POSTGRES_MFP.JCCLML_CHECKPOINT;1
                    64/72       16-SEP-2015 23:14:40.41  (RWED,RWED,RE,)
Total of 1 file, 64/72 blocks.
dba08 >
```

- All committed transaction will be replicated
- Restart may resend some transactions

View Restart Information

- jcc_lml_dump_checkpoint <LoaderName> <name> [type]
  - <LoaderName> – LoaderName for the checkpoint
  - <name> – checkpoint filename or target database
  - [type] – checkpoint stream type
    - LML_INTERNAL (Default)
    - OCI
    - RDB

JCC_LML_DUMP_CHECKPOINT Example

```
dba08 > jcc_lml_dump_checkpoint kwh_postgres
LOADER_CHKPT_POSTGRES_MFP.JCCLML_CHECKPOINT lml_internal

JCC LML Dump Checkpoint V03.04.04 (built 30-JUN-2014 09:28:28.13)

 -- Checkpoint restart information --

--- Parallel mode ---
Write Timestamp:     19-SEP-2015 15:25:36.50
LoaderName:          KWH_POSTGRES
Completion Flag:     S
Checkpoint Interval: 1
Input Data Source:   LML_CONT_KWH_POSTGRES
Last Transaction:
        Start Time:  18-SEP-2015 21:55:41.19
        Commit Time: 18-SEP-2015 21:55:50.29
        TSN:         90694
        LSN:         15165
        AERCP:       1-28-41-25-90694-90694
        RM TID:
```

Adding additional information

- Virtual Columns
  - ORIGINATING_DBKEY
  - LOADER_SEQUENCE_NUMBER
  - TRANSACTION_START_TIME
  - TRANSACTION_COMMIT_TIME
  - PID – Source transaction OpenVMS Process ID
  - Etc.
- Virtual Tables
  - Map Commit record to target table

Filtering Data

- FilterMap Example

```
FilterMap~EMPLOYEES~where \
COALESCE(BIRTHDAY,date vms'17-NOV-1858') between date vms'17-NOV-1858'
and date vms'31-DEC-9999'
```

- FilterMap statements must be after MapTable statement
- Looks like Rdb SQL
  - Processed using dynamic SQL and FilterMap DB
- Can only reference current row

FilterMap Database

- Rdb database specifically for evaluating filters
- Default is jcc_tool_data:<loadername>.rdb
- Logical Names
  - JCC_LOGMINER_LOADER_FILTER_DIR – specifies Filter DB Location
  - JCC_LOGMINER_LOADER_FILTER_NAME – overrides the <loadername>
- Automatically created when first needed

FilterMap Database

- Create your own if the defaults are not sufficient for your environment
- Example

```
CREATE DATABASE ALIAS filter_db FILENAME jcc_tool_data:<loadername>.rdb
        OPEN IS AUTOMATIC
        NUMBER OF USERS 33
        NUMBER OF CLUSTER NODES 1
        PAGE SIZE IS 4 BLOCKS
        BUFFER SIZE IS 4 BLOCKS
        GLOBAL BUFFERS ARE ENABLED
               (NUMBER IS 330,USER LIMIT IS 10,PAGE TRANSFER VIA DISK)
        DBKEY SCOPE IS TRANSACTION
        PRESTARTED TRANSACTIONS ARE OFF  --(or ON if you wish)
        DICTIONARY IS NOT REQUIRED
        SHARED MEMORY IS SYSTEM;
```

Transforming data

- Map one source table to multiple targets
- Map multiple source tables to one target table
- Turn specific values into NULLs
- Turn NULLs into values
- What if we could add a function to the FilterMap database?

MapResult

- In the Mythical Next Release (MNR)

```
MapTable~CANDIDATES~CANDIDATES,candidates~Replicate
MapColumn~CANDIDATES~LAST_NAME
MapColumn~CANDIDATES~FIRST_NAME
MapColumn~CANDIDATES~MIDDLE_INITIAL
MapColumn~CANDIDATES~CANDIDATE_STATUS
!Originating_DBKey_column_required_due_to_no_key.
MapColumn~CANDIDATES~originating_dbkey
MapKey~CANDIDATES~originating_dbkey
!
MapResult~CANDIDATES~text_dbkey~jcc$dbkey2text(originating_dbkey)
```

- jcc$dbkey2text
  - External function defined in FilterMap database
- Could use built-in Rdb functions

MapResult Example

- Rdb source database update:

```
SQL> update candidates set middle_initial = middle_initial;
3 rows updated
SQL> commit;
SQL>
```

- Postgres target database query:

```
keith=# select last_name, originating_dbkey, text_dbkey from candidates;
   last_name    | originating_dbkey |    text_dbkey
----------------+-------------------+-----------------
 Wilson         | 25332747945508864 | 90:634:0
 Schwartz       | 25332747945508865 | 90:634:1
 Boswick        | 25332747945508866 | 90:634:2
(3 rows)

keith=#
```

Multiple Update Threads

- Sometimes benefit in sending multiple threads of data to target
- Example

```
parallel~1~4~constrained
thread~startup~10
thread~shutdown~30
```

- 1 to 4 concurrent update threads
- Startup thread if data is available for more than 10 seconds while all threads are writing
- Shutdown thread if idle more than 30 seconds

Verifying Data in Target

Challenges comparing data in two databases

- Datatype representations
  - CHAR, VARCHAR – trailing spaces, character sets
  - Floating point – G-Float to IEEE precisions
  - Date, Time, Timestamp – precisions
- Data Volume – more rows, more time needed
- Freeze updates while doing a compare?
- Techniques beyond time allotted today

Additional targets

- Rdb targets are straightforward
- Oracle targets using Oracle Call Interface (OCI)
  - Install Oracle client on OpenVMS
  - Understand datatype differences
- JDBC Targets can be challenging
  - Install Java on OpenVMS
  - JDBC Driver files to OpenVMS & Reset Jar File characteristics
  - Specify class path
  - Specify appropriate connection string
  - Understand datatype differences

Summary

- Discussed what the JCC LogMiner Loader accomplishes
- Detailed example of setting up replication from Rdb on OpenVMS to Postgres on Linux
  - First loader setup is the hardest
  - Other sessions and targets use similar steps
- Hinted at additional LML options and capabilities

Acknowledgements

- Thanks to Rdb engineering for their support and counsel
- Thanks to our customers for sharing their experiences with the Loader

Join the Conversation

Join the worldwide Rdb community. Send mail to

OracleRdb-request@JCC.com with "SUBSCRIBE" in the body of the message.

Questions?

http://www.jcc.com/lml

At break,
please ask questions
and share ideas

Send your input and requests to jcc-lmloader@JCC.com