GoldenGate and ODI - A Perfect Match for Real-Time Data Warehousing

Michael Rainey, Principal Consultant, Rittman Mead
RMOUG Training Days, February 2013
About me...

• Michael Rainey, Principal Consultant, Rittman Mead
• Oracle Data Integration Specialist
  › Oracle Data Integrator 11g, Oracle GoldenGate 11g
• Writer for Rittman Mead Blog: http://www.rittmanmead.com/blog
• Email: michael.rainey@rittmanmead.com
• Twitter: @mRainey
About Rittman Mead...

• Oracle BI&DW Experts
  ‣ Consulting, Training and Support
  ‣ Global Company, Offices in the US, UK, India and Australia
  ‣ 1 Oracle ACE Director, 2 Oracle ACE’s
• Longest-running Oracle (BI) Blog
• Voted UKOUG BI Partner of the Year 2008, 2010, 2011
• Oracle Community Advocates
  ‣ Frequent speakers at user groups and conferences
  ‣ 36 Sessions at Oracle Open World since 2010
• Twitter : @rittmanmead
About this presentation...

• Real-time data warehousing is becoming standard across many organizations
• Oracle’s Next-Generation Reference Data Warehouse Architecture
  ‣ Staging, Foundation, and Access and Performance Layers
• Implementation of real-time data warehouse
  ‣ Oracle GoldenGate - replication technology
  ‣ Oracle Data Integrator - data integration & ETL
  ‣ GoldenGate and ODI integration
• Real-time ETL
  ‣ Solutions to common challenges
Staging Layer

- Staging schema stores exact copy of Source data
  - Tables built with no key constraints
- Typically loaded using change data capture
  - Oracle Streams/CDC, Oracle GoldenGate, etc
- Limited load on the Source database
Foundation Layer

- Similar to Staging, schema built to match Source
  ‣ Additional columns added for tracking historical data
    - SCN (system change number)
    - Commit timestamp
    - DML type (insert, update, delete)
- Append all transactional history from Source
  ‣ Allows drill-through from report for historical data
  ‣ Rapid development of new business requirements
  ‣ Additional facts and dimensions can be loaded as of the same initial load date
Access and Performance Layer

- Optimized for reporting access and performance
  - Dimensional models
  - Summary and aggregate data
  - Reporting tools can easily report against star schema
- Loaded using ETL tool such as OWB or ODI
  - Transformations use data from Staging tables
- History not required in denormalized star schema
Oracle GoldenGate

• Oracle’s standard tool for data replication
• Provides log-based capture, distribution, and delivery of committed transactions in real-time
  ‣ Sub-second replication time
  ‣ Minimal impact to source and target systems
  ‣ Utilizes platform independent universal data format
• Replication of data between heterogeneous systems
  ‣ Handles source and target column differences
• Uni-directional or bi-directional replication
• Easy to deploy - simple configuration of parameter files
Oracle Data Integrator 11g

- Oracle’s strategic product for data integration
- Supports batch, event-driven, and real-time integration
- Uses ELT (Extract, Load, Transform) approach
  - No middle ETL engine necessary
  - Utilizes power of target database to perform transformations
- Supports heterogeneous data sources
  - Oracle, SQL Server, XML, flat-file, MySQL, DB2...
- Declarative design - separation of business and technical integration
- Data integrity controls create a “data firewall”
- Extensible through “Knowledge Modules”
Oracle Data Integrator Journalizing (CDC)

• Change Data Capture (CDC)
  ‣ Identify, capture, and deliver changes made to data in the source database

• Oracle Data Integrator CDC delivered through Journalizing
  ‣ Journalizing Knowledge Module (JKM) performs setup and creates infrastructure

• ODI CDC Framework
  ‣ Journals - tables (J$) hold references to the changed records and the change type (insert/update/delete)
  ‣ Capture processes - captures changed data from source datastores
    - Database specific programs to retrieve log data from data server log files (Ex: Oracle GoldenGate)
  ‣ Subscribers - entities that use the changed data tracked on a datastore or consistent set
    - Data purged from journals after all subscribers have consumed changed data
  ‣ Journalizing views - (JV$, JV$D) provides access to changed data, used by IKM’s and LKM’s
GoldenGate and ODI Integration

• **JKM Oracle to Oracle Consistent (OGG) Knowledge Module**
  ‣ Delivered with ODI
  ‣ ODI Metadata used to generate GoldenGate parameter files (extract, datapump, replicat)
  ‣ ReadMe.txt file created with instructions
• **ODI CDC Framework generated**
  ‣ GoldenGate replicates to Staging table and J$ (journal) table
  ‣ J$ loaded using INSERTALLRECORDS option
• **Journalized data used to load downstream tables**
  ‣ ODI change views handle intra-commit transactions
**Staging Layer Load**

- GoldenGate replication is setup and configured to keep the Staging schema in sync with the Source
- All committed changes loaded to Staging in real-time
Foundation Layer Load

- Standard process was to load incremental changes from Staging to Foundation
  - Requires extra set of mappings to process data
  - Increases latency of real-time data warehouse load
- GoldenGate will load Foundation directly
  - Improved technology
  - Implementation experience
  - Reduced overall data warehouse load time
- A simple customization to the JKM will allow the generation of source to Foundation GoldenGate parameter files
  - Use INSERTALLRECORDS option for storing transactional history
Set the Journalizing Knowledge Module Options - Staging

Datastore

Model

Wednesday, February 13, 13
Journalizing Knowledge Module Options

- **LOCAL_TEMP_DIR**: Local path for generated parameter files
- **SRC_LSCHEMA**: Source Logical Schema
- **SRC_DB_USER**: Source GoldenGate user
- **SRC_DB_PASSWORD**: Source GoldenGate password
- **SRC_OGG_PATH**: Source GoldenGate install path
- **SRC_SETUP_OGG PROCESSES**: Setup extract files if true
- **STG_HOSTNAME**: Target server hostname
- **STG_MANAGER_PORT**: Target GoldenGate install port
- **STG_OGG_OBJECT_GROUP**: Replicat file name
- **STG_OGG_PATH**: Target GoldenGate install path
- **ENABLE_ODI_CDC**: Setup the ODI CDC framework if true
- **STG_OGG TRACK_HISTORY**: Custom option - store history
Configure GoldenGate

This readme.txt file provides a detail of the different steps to be executed manually in order to complete the setup of the CECC using OGG.

**UPLOAD FILES TO SOURCE MACHINE**
The files for the source machine have been generated in `/u01/temp/O01S_to_O01T1/src/
These files must be copied on the source machine, to the following location `/u01/ogg` using the same directory structure.

For instance, files contained in `/u01/temp/O01S_to_O01T1/src/dirprm` should end up in `/u01/ogg/dirprm`

**UPLOAD FILES TO STAGING MACHINE**
The files for the staging machine have been generated in `/u01/temp/O01S_to_O01T1/stg`
These files must be copied on the target machine, to the following location `/u01/ogg_stg` using the same directory structure.

For instance, files contained in `/u01/temp/O01S_to_O01T1/stg/dirprm` should end up in `/u01/ogg_stg/dirprm`

**RUN THE SOURCE OBEY FILE**
Connect to the source machine using a command line tool such as `cmd` on Windows or `sh` on UNIX/Linux and execute the following command:

```
/u01/ogg/ggscsi paramfile /u01/ogg/diroby/00ISS.oby
```

**GENERATE THE DEFINITION FILE**
Set the Journalizing Knowledge Module Parameters - Foundation

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>VALIDATE</td>
<td>&lt;Default&gt;-False</td>
</tr>
<tr>
<td>LOCAL_TEMP_DIR</td>
<td>/u/01/temp</td>
</tr>
<tr>
<td>NB_APPEND_PROCESS</td>
<td>&lt;Default&gt;-2</td>
</tr>
<tr>
<td>TRAIL_FILE_SIZE</td>
<td>&lt;Default&gt;-500</td>
</tr>
<tr>
<td>SRC_OCC_OBJECT_GROUP</td>
<td>&lt;Default&gt;-ODIR</td>
</tr>
<tr>
<td>SRC_SCHEMA</td>
<td>ORCL_NFL_STATS_SRC</td>
</tr>
<tr>
<td>SRC_DBUSER</td>
<td>ggs_owner</td>
</tr>
<tr>
<td>SRC_DB_PASSWORD</td>
<td>ggs_owner</td>
</tr>
<tr>
<td>SRC_OCC_PATH</td>
<td>/u/01/ggs</td>
</tr>
<tr>
<td>SRC_OCCE,Object_Group</td>
<td>&lt;Default&gt;-ODIR</td>
</tr>
<tr>
<td>STG_OCCE_OBJECT_GROUP</td>
<td>&lt;Default&gt;-true</td>
</tr>
<tr>
<td>STG_HOSTNAME</td>
<td>&lt;Default&gt;-localhost</td>
</tr>
<tr>
<td>STG_MANAGER_PORT</td>
<td>7310</td>
</tr>
<tr>
<td>STG_OCCE_PATH</td>
<td>/u/01/ggs_farr</td>
</tr>
<tr>
<td>STG_OCCE_OBJECT_GROUP</td>
<td>&lt;Default&gt;-true</td>
</tr>
<tr>
<td>COMPATIBLE</td>
<td>&lt;Default&gt;-22</td>
</tr>
<tr>
<td>CHECKPOINT_TABLE_NAME</td>
<td>&lt;Default&gt;-0000000001</td>
</tr>
<tr>
<td>USE_OCCE_F4R_INIT</td>
<td>&lt;Default&gt;-false</td>
</tr>
<tr>
<td>CLUSTER_DATABASE_INSTANCES</td>
<td>&lt;Default&gt;-0</td>
</tr>
<tr>
<td>CHARSET_ENCODING</td>
<td>&lt;Default&gt;-ISO8859-1</td>
</tr>
<tr>
<td>USE_QUALIFY_CONNECTION_TO_4TARNG</td>
<td>&lt;Default&gt;-false</td>
</tr>
</tbody>
</table>

```sql
mapNFL_STATS.GAME_PLAY_DETAILS, TARGET EDW.FND.GAME_PLAY_DETAILS,
KEYCOLUMNS (GAME_DATE, HOME_TEAM, AWAY_TEAM, QUARTER, TIME, PLAY_TYPE,
PLAYER, (EDW_SCN INSERTALLRECORDS)
COLMAP (USEDEFAULTS,
EDW_COMMIT_DATE = @GETENV("GGHEADER", "COMMITTIMESTAMP"),
EDW_SCN = @GETENV("TRANSACTION", "CSN"),
EDW_DML_TYPE = @GETENV("GGHEADER", "DPTYPE"),
} ;
```
Initial Load of Target and GoldenGate Startup

- Run initial load of source data before starting replication
  - Recommended tools: Oracle Datapump, Oracle Export/Import, DBLink
- Example initial load and GoldenGate startup process:
  - Follow instructions to setup the GoldenGate parameter files
  - Start the GoldenGate extract and datapump processes
  - Run the initial load (Ex: Oracle Datapump as of SCN)
  - Once the initial load has completed, start the GoldenGate replicat process after the initial load SCN
    - GGSCI >start replicat ODIT1T afterCSN 123456
  - Handling data collisions should not be necessary
Access and Performance Layer Load

• Moving change rows through to the star schema
• Common challenges
  ‣ Slow performance
  ‣ Missed joins between change tables
  ‣ Orphaned child records
• Several solutions
  ‣ Micro-batch ETL
  ‣ Multiple ODI Interfaces to handle Parent-Child relationship
  ‣ “Subscription Views” to provide a consistent view of the data - combining change rows with current rows
Micro-batch ETL

- Use an additional “Intermediate Staging” area to capture changes
  - Allows large source tables to load into the data warehouse faster
- Near real-time data load from Staging to Intermediate Staging
- Micro-batch ETL process
  - Build Interface to move change rows from J$ table and Staging to Intermediate Staging
  - Build and execute transformations as usual
Micro-batch ODI Interface - Design
**Micro-batch ODI Interface - Execution**

```
insert /*+ APPEND */ into EOD.Stg.1$OFFENSE_PLAYMAKERS
(
    TEAM,
    PLAYER_NAME,
    POSITION,
    NUMBER_OF_PLAYS,
    INO_UPDATE
) 
select
    OFFENSE_PLAYMAKERS.TEAM,
    OFFENSE_PLAYMAKERS.PLAYER_NAME,
    OFFENSE_PLAYMAKERS.POSITION,
    OFFENSE_PLAYMAKERS.NUMBER_OF_PLAYS,
    JRN_FLAG INO_UPDATE
from EOD.Stg.1$OFFENSE_PLAYMAKERS
where (i=1)
And JRN_SUBSCRIPTOR = 'ODI'
```

![Diagram showing execution flow: Extend & Lock, LOAD GAME_PLAY DETAILS micro-batch, LOAD OFFENSE_PLAYMAKERS micro-batch, LOAD ROSTER micro-batch, Unlock & Purge]
Parent-child Tables

Parent

Child

left outer join

left outer join

T : +44 (0) 8446 697 995  E : enquiries@rittmanmead.com  W: www.rittmanmead.com

Wednesday, February 13, 13
Parent-child Tables

- Capture changes to the parent table and child table
  - Handles orphaned child records
- Create two ODI Interfaces
  - Interface 1: Parent table with Journalizing enabled, full child table
  - Interface 2: Child table with Journalizing enabled, full parent table
  - Both have the same logic, column mappings, etc
  - Consistent Journalizing must be used to ensure a consistent dataset
Parent-child Tables - Design

- Journalized Data only: check box
- Use Temporary Interface as Derived Table (Sub-Select): check box
- Alias: P
- Order: 1
- Context: <Execution Context>

- Source Properties
- Journalized Data only: check box
- Use Temporary Interface as Derived Table (Sub-Select): check box
- Alias: G
- Order: 2
- Context: <Execution Context>
- Partition/Sub-Partition: <Next>

Extend & Link
Parent Offensive Plays load
Child Offensive Plays load
Unlock & Purge

T: +44 (0) 8446 697 995  E: enquiries@rittmanmead.com  W: www.rittmanmead.com
Subscription Views

- Always returns a consistent set of data
- Reduces the number of mappings
  - Parent-child issues resolved
  - Intermediate staging unnecessary
- Create a view for each Staging table
  - Left join all rows from change set (JV$D view)
  - Resulting dataset combines change rows and existing rows
- ETL developer can choose between current records or change dataset only
  - CURRENT_IND = ‘Y’ - all new and existing current rows
  - STAGE_IND = ‘Y’ - all new rows from the change table
Subscription Views - SQL Code

```sql
CREATE OR REPLACE VIEW SUB_OFFENSE_PLAYMAKERS AS
SELECT
  NVL(C.TEAM, T.TEAM) TEAM,
  NVL(C.PLAYER_NAME, T.PLAYER_NAME) PLAYER_NAME,
  NVL(C.POSITION, T.POSITION) POSITION,
  NVL(C.NUMBER_OFPLAYS, T.NUMBER_OFPLAYS) NUMBER_OFPLAYS,
  E.ROLE, E.LANGUAGE TYPE,
  NVL(C.STAGE_IND, T.STAGE_IND) STAGE_IND,
  NVL(C.CURRENT_IND, T.CURRENT_IND) CURRENT_IND
FROM
  (SELECT
   A.TEAM,
   A.PLAYER_NAME,
   A.POSITION,
   A.NUMBER_OFPLAYS,
   NULL EMPL_DL_TYPE,
   'Y' STAGE_IND,
   'Y' CURRENT_IND
   FROM EMPL_STG.OFFENSE_PLAYMAKERS A ) S
LEFT OUTER JOIN
  (SELECT
   C.TEAM,
   C.PLAYER_NAME,
   C.POSITION,
   C.NUMBER_OFPLAYS,
   C.JUN_FLAG EMPL_DL_TYPE,
   'Y' STAGE_IND,
   'Y' CURRENT_IND
   FROM EMPL_STG.JUNOFFENSE_PLAYMAKERS C ) C
ON S.TEAM = C.TEAM
AND S.PLAYER_NAME = C.PLAYER_NAME
```
Subscription Views - Design and Execution
**In summary...**

- Real-time data warehousing is now a standard across many organizations.
- Oracle’s Next-Generation Reference Data Warehouse Architecture provides a great structure for implementation.
- GoldenGate and ODI the delivery mechanism, and integrated are the perfect match for real-time data warehousing.
- Several solutions to real-time ETL challenges were provided:
  - Micro-batch loading, parent-child relationships, and subscription views.

More information can be found at [http://www.rittmanmead.com](http://www.rittmanmead.com).
Contact us at info@rittmanmead.com or michael.rainey@rittmanmead.com.
Follow-us on Twitter (@rittmanmead & @mRainey) or Facebook (facebook.com/rittmanmead).
Goldengate and ODI - A Perfect Match for Real-Time Data Warehousing

Michael Rainey, Principal Consultant, Rittman Mead
RMOUG Training Days, February 2013

T: +44 (0) 8446 697 995  E: enquiries@rittmanmead.com  W: www.rittmanmead.com