xtime
Transforming Service. Driving Success.

Architecting Multi-tenancy in Cloud using OBIEE and Oracle DB

Mayank Srivastava I Chief Architect

Hanan Hit I Database Architect

NoCOUG Autumn 2013

11/21/2013

1400 Bridge Parkway, Suite 203 | Redwood Shores, CA 94605 | www.xtime.com

OBIEE Multi-tenant Architecture

xtime

- Schema and RPD Flexibility
- Customizing Tenant Experience
- Customizing Tenant Performance
- Multi-tenant Security

OBIEE Multi-tenant Architecture

xtime

- Schema Flexibility
  - Flexible Columns in the schema for
    - Date
    - Text
    - Numeric Column Pair (value, Unit of Measure)
    - Currency Column Pair (value, Unit of Measure)
    - Boolean
- RPD Flexibility
  - Tenant based aggregations on the same column
    - Flexible Field as a gate
  - Externalize Flexible Column Names
    - Metadata table maps flex fields with tenants and business names
  - Dynamic Association of Slowly Changing Hierarchies
    - Create joins in RPD; not in database

OBIEE Multi-tenant Architecture

xtime

- Customizing Tenant Experience
  - Role based section switching
    - Display sections in dashboard based on roles from session variable
  - Role based defaults on prompts
    - Create multiple prompts with default values to be activated based on roles
  - Tenant based externalization of column names
    - Metadata table maps flex fields with tenants and business names
  - Tenant based custom skin and style sheets
    - Deploy S_TenantName (stylesheet) and SK_TenantName (skins) in weblogic by copying existing S_BLAFP and SK_BLAFP
    - Add tags in intanceconfig.xml
    - Create table with username, S_TenantName, SK_TenantName
    - Define init block with target ‘SKIN’ and ‘STYLE’ session variables

Performance

- Customizing Tenant Performance
  - Tenant based partitioning in RPD
    - Create separate partitions for premium tenants
  - Tenant based caching
    - After data load, purge cache and seed the cache with data of premium tenants
  - Tenant based connections
    - Create multiple DB connections and assign in Init Block based on tenant
    - Assign high performing DB connections to premium tenants
  - Tenant based row control
    - Control rows based on tenant in the RPD based on the roles
    - Control number of rows in a report based on Tenant using TopN expression

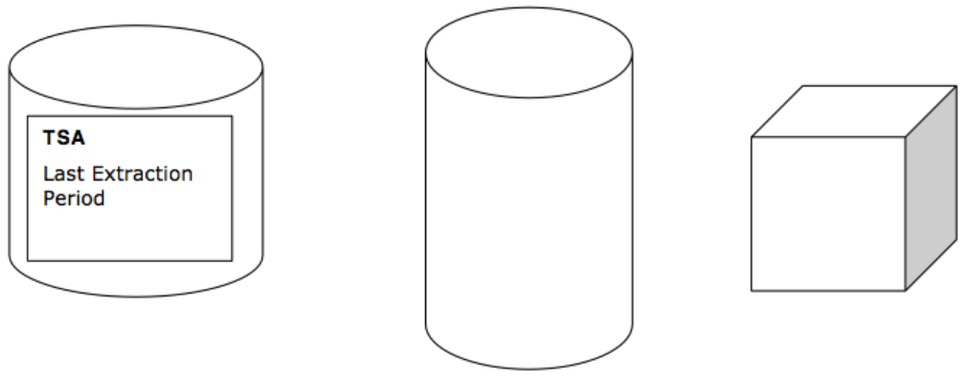
OBIEE Multi-tenant Architecture

xtime

- Multi-tenant Security
  - Data Level Security based on user scope
    - Restrict data fetched based on scope in the init block
  - Catalog Security based on user role
    - Restrict objects based on the role in the init block
  - Report Section Security based on user role
    - Control sections in a dashboard based on tenant
  - Hierarchy Node Security based on user scope
    - Limit data access at different levels of hierarchy based on scope
  - Multiple Security within a Dashboard
    - Use direct SQL in report to get benchmarking data in the same dashboard

Data Warehouse Loading Schema

General warehouse Database Loading



FULL ETL

xtime

Incremental ETL

Incremental ETL Cont.

Incremental ETL – Partition Exchange

Incremental ETL – Views Creation

Håkan factor ???

xtime

1. Created a new table with the same base table structure, loaded the data and created the matching indexes.
2. You try to issue “ALTER TABLE … EXCHANGE PARTITION … WITH TABLE … INCLUDING INDEXES” but getting that:

14642. 00000 - "Bitmap index mismatch for tables in ALTER TABLE EXCHANGE PARTITION"

*Cause: The two tables in the EXCHANGE have usable bitmap indexes, and the INCLUDING INDEXES option has been specified and the tables have different hakan factors.

*Action: Perform the exchange with the EXCLUDING INDEXES option or alter bitmap indexes to be unusable.

3. So where is the Error?

Håkan factor ??? Cont.

1. The term is used to denote the maximum number of rows in a data block of a table.
2. This is relevant for bitmap indexes, because they represent each data block by a number of bits, one for each row in the block. So the Håkan factor is the number of bits allocated for each data block.
3. For partition exchange to work with bitmap indexes, the Håkan factor of the two tables must be the same.
4. The way to identify it is by executing the below query:

```
SQL> Select Object_Name,To_Char(Spare1,'0xxxxx')
Spare1_Hex,Bitand(Spare1,32767) Hakan_Factor
From Dba_Objects O,Sys.Tab$ T Where
Object_Name ='<table name>' and OWNER = '<owner name>'
And Object_Type='TABLE' and o.object_id=t.obj#;
```

Håkan factor ??? Cont.

- Source table - partitioned

OBJECT_NAME	*SPARE1_*	*HAKAN_FACTOR*
DW_RO_FACT	0102e0	736

- Dest Table - Non partitioned

OBJECT_NAME	*SPARE1_*	*HAKAN_FACTOR*
CURR_DW_RO_FACT_ADD	00008e	142

Håkan factor ??? Cont.

xtime

1. The solution is: Use Oracle event 14529.
2. Make sure that event 14529 is set and unset at the session FOR each table.
3. Create the table as SELECT * from the original table.

```
v_sql_stmt := ' ALTER SESSION SET EVENTS "14529 TRACE NAME CONTEXT FOREVER,
LEVEL 1"'  ;   -- See note ID 1201195.1
Execute Immediate V_Sql_Stmt;
V_Sql_Stmt := ' CREATE TABLE '  II V_Schema_Name II '.' II 'CURR_' II I II '_' II
Factrec.Fact_Table_Name II ' Tablespace ' II V_Tab_Tbs II '   AS SELECT * FROM ' II
V_Schema_Name II '.' II  Factrec.Fact_Table_Name  II  '  WHERE 0=1' ;
Execute Immediate V_Sql_Stmt;
v_sql_stmt := ' ALTER SESSION SET EVENTS "14529 TRACE NAME CONTEXT OFF"'  ;
Execute Immediate V_Sql_Stmt;
```

12c Migartion

12c migration – quick lessons learned

xtime

1. If using Container & PLUGGABLE DATABASE's make sure to use/modify the scripts to use SQL*Net instead of BEQ – No ORACLE_SID for the Pluggable Database.
2. Most of the stuff works the same – No issues with OBIEE and Informatica Power Center 9.5.
3. Database files for Pluggable Database are created by default under the CONTAINTER_DATABASE_SID/<Pluggable Database *sid*>.
4. SYSTEM/SYSAUX and Temp are created for container as well for Pluggable Database.
5. UNDOTBS doesn't exists at the Pluggable Database DB – Only at the container DB.
6. Operations such as create pfile (for example) are only allowed at the container level.
7. Always use "show con_name" in Sql*Plus to make sure where you are at.

ZFS – Options for Recovery Using Snapshot Copies Without Backup Mode

Options for Recovery

Crash-Consistent Snapshot Copy

xtime

- On the Fly Snapshot

```
SQL> startup mount;
ORACLE instance started.
Total System Global Area 5.7322E+10 bytes
…
Database mounted.

SQL> select * from v$recover_file;
no rows selected
SQL> alter database open;
Database altered.
```

xtime

On the Fly Snapshot

```
sudo zfs snap zdbpool/oracle11@FullDBRecovery_ZeroDataLoss
$ sudo zfs list -t snap
NAME USED AVAIL REFER MOUNTPOINT
zdbpool/oracle11@BlankDB
zdbpool/oracle11@FullDB
zdbpool/oracle11@BackupMode zdbpool/
oracle11@FullDBRecovery_CGsnap_Offline 15.8M - 2.41T - zdbpool/
oracle11@FullDBRecovery_CGsnap_Online 15.2M - 2.41T - zdbpool/
oracle11@FullDBRecovery_ZeroDataLoss 39.1M - 2.41T -
```

xtime

SHUTDOWN IMMEDIATE

- cp *.log /home/oraprd/hhit/Saved_Redo_Control
- cp *.ctl /home/oraprd/hhit/Saved_Redo_Control

$ sudo zfs rollback zdbpool/oracle11@FullDBRecovery_ZeroDa

Recover the Current redo and control files

- cp *.log /zdbpool/oracle11/oradata/bisnap/bisnap
- cp *.ctl /zdbpool/oracle11/oradata/bisnap/bisnap

STARTUP MOUNT

SQL> select * from v$recover_file

Will show files that should be recovered

```
SQL> RECOVER AUTOMATIC DATABASE;
Media recovery complete.
SQL> select * from v$recover_file;
no rows selected
SQL> alter database open;
Database altered.
```

POINT In Time database Recovery

- $ sudo zfs snap zdbpool/oracle11@FullDBRecovery_PIT_Recovery
- *SHUTDOWN IMMEDIATE*
- cp *.log /home/oraprd/hhit/Saved_Redo_Control
- cp *.ctl /home/oraprd/hhit/Saved_Redo_Control
- Rollback Snapshot
- $ sudo zfs rollback zdbpool/oracle11@FullDBRecovery_PIT_Recovery
- Recover the Current redo and control files
- cp *.log /zdbpool/oracle11/oradata/bisnap/bisnap
- cp *.ctl /zdbpool/oracle11/oradata/bisnap/bisnap

POINT In Time database Recovery – Cont.

xtime

- *STARTUP MOUNT*

Identify the minimum database SCN required for to be consistent

File 80 absolute fuzzy scn = 0
Minimum PITR SCN = 2637341

PL/SQL procedure successfully completed.

Elapsed: 05:50:23.77

Recover (Incomplete) to the minimum identified SCN

SQL> RECOVER AUTOMATIC DATABASE UNTIL CHANGE 2637341;

Media recovery complete.

SQL> ALTER DATABASE OPEN RESETLOGS;

Database altered.