



COLLABORATE 15

TECHNOLOGY AND APPLICATIONS FORUM
FOR THE ORACLE COMMUNITY

Pluggable Databases : Save money and time by harnessing the power of ORACLE for database consolidation

Session ID#: 830

Prepared by:

John Larkin

Sr. Database Administrator, OCP
Advanced Database Services, LLC.



John.larkin5@Verizon.net



APRIL 12-16, 2015
MANDALAY BAY
RESORT & CASINO

#C15LV

REMINDER

Check in on the
COLLABORATE mobile app

Pluggable Databases : Save money and time cont'd

- Oracle DBA for 18 years. (28 yrs overall).
- OCP
- UNIX (Linux, Solaris, Aix, Windows)
- Multi-year project to migrate from VCS to RAC One-Node
- Major Financial, Chemical, Publishing and Retail industries.

- john.larkin5@verizon.net



Why do we want to use a Container Database?

Retain cost savings and efficiencies of shared resources in the next generation server environment.



COLLABORATE 15

TECHNOLOGY AND APPLICATIONS FORUM
FOR THE ORACLE COMMUNITY

Pluggable Databases : Agenda

- Consolidation Overview
- Multitenant concepts
 - Scope
 - Currency
 - Services
 - Connections
 - Commonality
- Installation
- Physical Structure
- Migration – Get Plugged In
- Gotcha's



Pluggable Databases : Save money and time cont'd

- Consolidation
 - Prior methods worked at different levels
 - Server
 - Schema
 - issues
 - Oracle Database 12c
 - Integrated into the database
 - Decreased cost
 - Higher density
 - Shared memory
 - Shared processes



Pluggable Databases : Save money and time cont'd

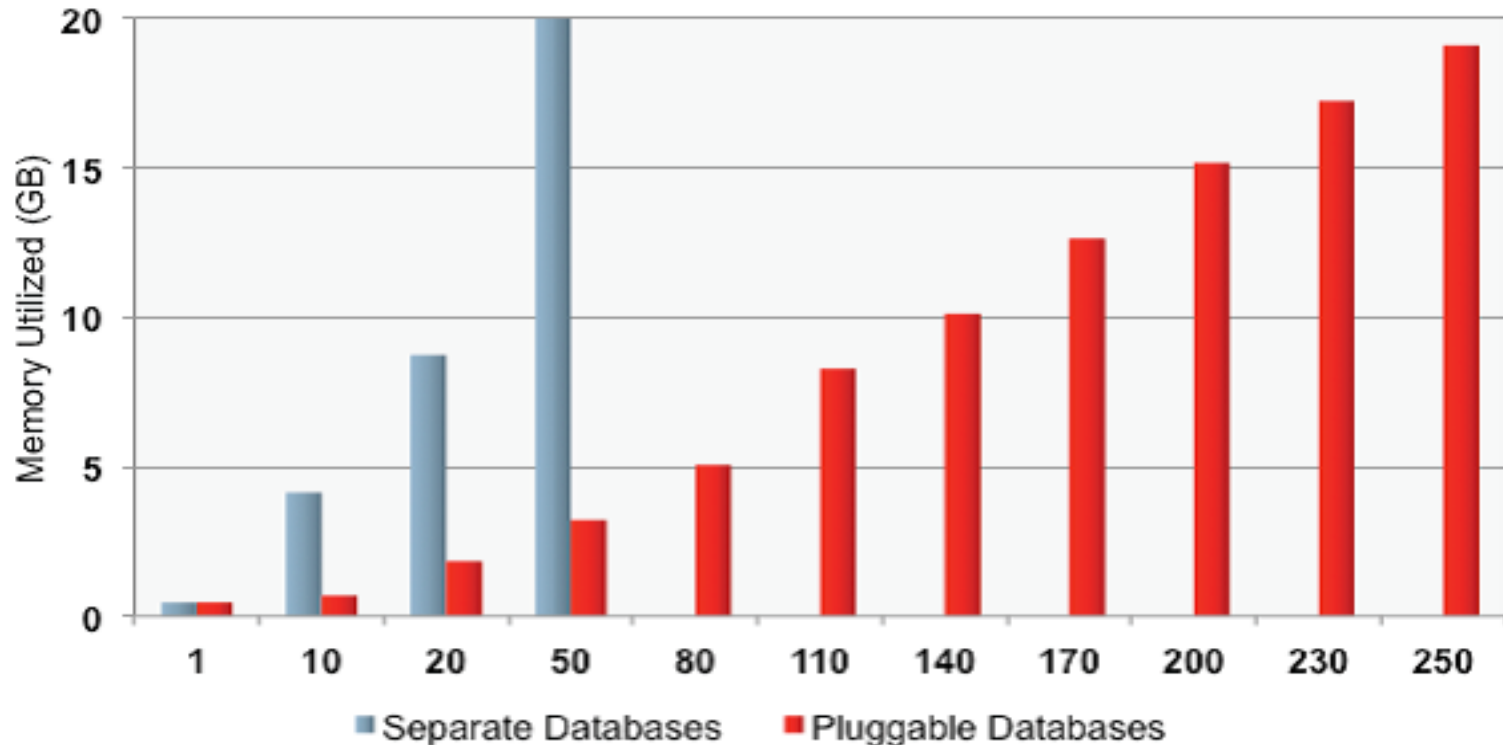


Figure 2: OLTP Benchmark Comparison Only 3 GB of memory vs. 20 GB memory used for 50 databases. Multitenant architecture scaled to over 250 DBs while separate database



Pluggable Databases : Save money and time cont'd

- Multitenant Container Database Concepts
 - The CDB – the backbone of Oracle’s multitenant architecture
 - holds the Containers
 - Root Container
 - Seed Pluggable database
 - Pluggable database – 0 or more
 - CDB is the instance
 - Demo
 - @d010



Pluggable Databases : Save money and time cont'd

CDB is the instance – cont'd

- (sqlplus / as sysdba)

Instance	VERSION	User	Dbname
-----	-----	-----	-----
orcl12a	12.1.0.2.0	SYS	ORCL12A

CON_NAME	CON_ID	(CONTAINER information)
-----	-----	
CDB\$ROOT	1	








(alter session set container=pdborcl12a1)

CON_NAME	CON_ID	(PDB information)
-----	-----	
PDBORCL12A1	3	



Pluggable Databases : Save money and time cont'd

CDB is the instance – cont'd (Windows Services window)

Name	Description	Status	Startup Type
 Oraclec_app_oracle_product_12102~1.0ConfigurationManager		Running	Automatic
 OracleJobSchedulerORCL12A			Disabled
 OracleOraDB12Home1MTSRecoveryService			Automatic
 OracleOraDB12Home1TNSListener		Running	Automatic
 OracleRemExecServiceV2		Running	Manual
 OracleServiceORCL12A		Running	Automatic
 OracleVssWriterORCL12A		Running	Automatic



Pluggable Databases : Save money and time cont'd

Some important Terms

- CDB – multitenant container database – the top level.
 - A database that holds the containers.
- Container - all within a CDB
 - collection of schemas, objects, related structures in a CDB
 - Appears to a client (user or application) as a separate database.
 - Unique ID and name within a CDB
 - Root and all PDB's.
 - Indistinguishable from non-CDB (12c and all pre-12c) to users.
 - Unless... you're a DBA.
 - Isolate and insulate



Pluggable Databases : Save money and time cont'd

- The ROOT container
 - CDB\$ROOT
 - Only 1 per CDB
 - All PDB's belong to it.
 - No user data
 - Common user (prefix/suffix modifiable)



Pluggable Databases : Save money and time cont'd

- PDB – Pluggable database, a container
 - Compatibility Guarantee
 - A PDB is fully compatible with a non-CDB
 - Application to run without code changes and deliver the same results
 - Init.ora changes might be needed
 - User-created
 - Owned by SYS common user
 - Looks like a standalone database
 - Segregate application data and code
 - Migrations
 - from one CDB to another
 - Upgrades



Pluggable Databases : Save money and time cont'd

- PDB cont'd
 - Services
 - PDB accessed by a service name that is the same as the PDB name
 - PDB name must conform to service naming standards
 - SCOPE
 - Name resolution - dictionary of the container where you're connected
 - Dictionary is horizontally partitioned.
 - Schemas in different databases
 - Independent, different datastore
 - » Even if owned by a common user
 - Behaves like a non-CDB



Pluggable Databases : Save money and time cont'd

- Data Dictionary Architecture/
 - non-CDB dictionary
 - a mixture of Oracle-supplied and user-created objects
 - CDB Dictionaries appear to be separated by container
 - Data dictionary metadata is split between the root and the PDB's
 - Views show different row counts in each container.
 - Keeps the oracle-owned and user-created objects separate.
 - ROOT
 - Oracle-owned objects
 - PDB
 - Unique user-created data
 - Pointers to ROOT dictionary



Pluggable Databases : Save money and time cont'd

- Data Dictionary Architecture cont'd
 - Reduce unnecessary duplication
 - Efficient upgrade path
 - i.e. DBMS_SCHEDULER package only in CDB\$ROOT
- Dictionary Separation provided by Links
- Metadata Links
 - Managed automatically, not user-modifiable
 - maintain the metadata about dictionary objects in the root
 - **column definitions** for Oracle-owned table exist only in the ROOT
 - **Metadata link** in the PDB points to the **definition** in the ROOT
 - i.e. OBJ\$ - defined in ROOT, data in PDB
 - User-defined objects reside completely in the PDB.(container)
 - Includes init.ora



Pluggable Databases : Save money and time cont'd

- Metadata Links cont'd
- Object Links - The other side of the coin
 - Store non-root container data for some special objects in the ROOT
 - AWR data (DBA_HIST_ACTIVE_SESSION)
 - Available to all containers.
- Container Data Objects (CDO_(my term))
 - Tables/views - data from multiple containers and/or the CDB
 - built-in - restricts data based on common user permissions
 - Oracle-Supplied views - V\$ and CDB_ are examples of CDO's
 - CON_ID column – determine source
 - CDB_* data returned based on which container you're connected to:
 - From ROOT – query metadata across all containers
 - From PDB – returns data only from that container



Pluggable Databases : Save money and time cont'd

■ Container Data Objects cont'd

- | <u>Container ID</u> | <u>Maps To</u> |
|---------------------|----------------------------|
| 0 | The Whole CDB or a non-CDB |
| 1 | CDB\$ROOT |
| 2 | PDB\$SEED |
| 3+ | User-created PDB |

- DEMO
@d020



Pluggable Databases :

Save money and time cont'd

- CDB_ vs DBA_

- ROOT: PDBORCL12a1:

- select count(*)

- from **CDB**_tables

- COUNT(*)

- -----

- 4747 2409

- select count(*)

- from **DBA**_tables

- COUNT(*)

- -----

- 2338 2409



Pluggable Databases : Save money and time cont'd

- Current Container
- The container in which the current session is running.
 - can be in the root, for common users
 - a PDB.
- A session has only one current container at any point in time.
 - name resolution /privilege authorization – curr. container's dict.
- Cross container operation
 - DDL statement that affects the CDB, multiple containers, entities in common containers or a container different than the issuing user's current container.
 - include database recovery and common user modifications.
- DEMO - files
 - @d030



Pluggable Databases : Save money and time cont'd

- Services and Connections
- connect to a PDB using a service
 - starts a session in a PDB
 - current container - permanent for the lifetime of the session.



Pluggable Databases : Save money and time cont'd

■ Demo

- Standalone database
 - Upgrade to 12c
 - Other options, but more complicated
- Read-Only
- Describe the standalone db – generate xml manifest
 - `dbms_pdb.describe()`
 - Back it up w/ the datafiles – SCN in all must match
- In CDB:
 - `dbms_pdb.check_plug_compatibility`
 - query `pdb_plug_in_violations` for `ERRORS/WARNINGS(cdb$root)`
- Create `PLUGGABLE DATABASE pdb1` using `'c:\manifest.xml'`
- Run `noncdb_to_pdb.sql` in the NEW PDB.



Pluggable Databases : Save money and time cont'd

DEMO cont'd

- Create description/manifest

```
exec dbms_pdb.describe(pdb_descr_file=>  
                        'C:\Data\Collab2015\noncdb12b.xml');
```



COLLABORATE 15

TECHNOLOGY AND APPLICATIONS FORUM
FOR THE ORACLE COMMUNITY

Pluggable Databases : Save money and time cont'd

DEMO cont'd

DECLARE

```
compatibility CONSTANT VARCHAR2(3) :=
```

```
CASE DBMS_PDB.CHECK_PLUG_COMPATIBILITY(  
pdb_descr_file => 'C:\Data\Collab2015\noncdb12b.xml',  
pdb_name => 'PDBORCL12A2')
```

```
WHEN TRUE THEN 'Db to be PLUGGED in is COMPATIBLE'
```

```
ELSE 'Db to be PLUGGED in is *** NOT COMPATIBLE ***'
```

```
END;
```

```
BEGIN
```

```
DBMS_OUTPUT.PUT_LINE(compatibility);
```

```
END;
```



Pluggable Databases : Save money and time cont'd

DEMO cont'd

- Check for Errors (still in CDB\$ROOT)
 - SET lines 200
 - col message FOR a100
 - SET pages 100
 - SELECT name,cause,TYPE,message FROM PDB_PLUG_IN_VIOLATIONS WHERE name='PDBORCL12A2';



Pluggable Databases : Save money and time cont'd

- DEMO cont'd
 - DEMO
xx@d040



Pluggable Databases : Save money and time cont'd

- DEMO cont'd



Pluggable Databases : Save money and time cont'd

- Gotcha's
 - Windows user accounts
 - Standardization
 - Database options
 - Database version
 - Multiple ways to get there
 - Non-CDBs are deprecated in 12c
 - Oracle recommends using CDBs (single-tenant or multitenant)
 - New paradigm – it's coming, best to get used to it now
 - Noncdb_to_pdb.sql – errors out/closes sqlplus if not run from the PDB - duh.



Pluggable Databases : Save money and time cont'd

■ References

- Tanel Poder
- also see <http://blog.yannickjaquier.com/oracle/multitenant-standalone-to-pluggable-migration.html>



Please complete the session evaluation

We appreciate your feedback and insight

You may complete the session evaluation either on paper or online via the mobile app



COLLABORATE 15

TECHNOLOGY AND APPLICATIONS FORUM
FOR THE ORACLE COMMUNITY

Demo Scripts

```
-- --- DEMO START 0 ----- CDB/PDB
```

```
select  
INSTANCE_NAME, CON_ID,  
VERSION, status, logins, shutdown_pending,  
DATABASE_STATUS, ACTIVE_State  
from v$instance  
/
```

```
prompt CDB/PDB info  
COLUMN name FORMAT A30
```

```
prompt SERVICEt info:  
SELECT name, pdb  
FROM v$services  
ORDER BY name  
/
```



Demo Scripts

```
-- --- DEMO START 0 ----- CDB/PDB (cont'd)
```

```
prompt CURRENT CONTAINER info:
```

```
--col "Container" format a16
```

```
--col "Container_ID" format a16
```

```
col CON_NAME format a16
```

```
col CON_ID format a16
```

```
SELECT SYS_CONTEXT('USERENV', 'CON_NAME') "CON_NAME",  
SYS_CONTEXT('USERENV', 'CON_ID') "CON_ID"
```

```
FROM dual
```

```
/
```

```
-- dupe name - diff datatype
```

```
-- cdb_info.sql - almost the same as show pdbs
```

```
col CON_ID format 999999999999
```

```
prompt PDB info:
```

```
select
```

```
CON_ID,
```

```
--DBID,
```



Demo Scripts

```
-- --- DEMO START 0 ----- CDB/PDB (cont'd)
```

```
-- condollar - show all containers and id's
```

```
select CON_ID#, DBID  
from sys.CONTAINERS$  
/
```

```
-- cdatafile_info
```

```
-- cdatafile_info.sql
```

```
col file_name for a75
```

```
col TABLESPACE_NAME for a30
```

```
prompt CDB_data_files
```

```
--select * from cdb_data_files
```

```
select
```

```
CON_ID,
```

```
FILE_NAME,
```

```
FILE_ID,
```

```
Rpad(TABLESPACE_NAME, 30, ' ')
```

```
.) TABLESPACE_NAME
```



COLLABORATE 15

TECHNOLOGY AND APPLICATIONS FORUM
FOR THE ORACLE COMMUNITY