**Multitenancy in Oracle 12c database: Pluggable Databases**

Abstract

Oracle 12c database is known for promoting the architectural concept of multitenancy. Multitenancy is defined as the ability of multiple databases to run under one instance of Oracle database software. The new multitenancy feature allows container database views for pluggable databases. This allows a user to see and manage the root container out of which the pluggable databases are tenanted out. There are several advantages in using multitenant databases which provide increased scalability, optimized resource utilization, improved performance, rapid provisioning along with consolidated management and patching.

Traditional databases have usually required diligent maintenance and provisioning actions undertaken by the DBA. Previously in any data intensive project that utilized multiple databases the DBA would have to create a separate instance for each database using Oracle 11g. Each instance would spawn its own background processes and metadata which are memory intensive and resource consuming. Each instance of the Oracle 11 g database would have to be managed and special attention and time would have to be placed on the filesystem structures and variables across different operating systems. These maintenance activities would require time consuming and meticulous attention by the DBA in order to upkeep thousands of production databases. However, Oracle 12c’s new multitenant features helps tackle this problem as it recommends up to 250 tenant databases to be run out of a single container database. (Garret, 2) The container database has only one copy of Oracles metadata. The tenant database contain individual copies of the customer’s metadata. This greatly simplifies database management and administration.

Beyond promoting elastic scalability and enhanced performance the multitenant architecture also promises stronger security for users. Oracle 12c provides a consolidated point of access and control for user. From this central container hub distributing security patches out to the individual tenants makes the new database architecture more secure.

Analysis

Prior to Oracle 12c, traditional IT environments were configured to reflect a database coupled with a server and dedicated to an application. However in this configuration template server utilization peaked only up to 10 percent. (Oracle Tech Writer, 27) Whenever the database performed intensive work a vast majority of server resources were unutilized and frozen. The cost of maintaining the physical infrastructure in this model became cost prohibitive and cumbersome over time. As this configuration template was scaled up it only contributed to rising costs of maintenance for the physical hardware and the associated databases. Since the newer Oracle 12c multitenant databases were released we can now reduce physical resource consumption and promote efficient database management, scaling and server utilization.

Multitenancy specifies one container database and several pluggable databases operating under it. This offering allows for one set of processes to moderate several 12c databases instead of the individual memory requirements of multiple separate oracle 11g instances. Multiple tenants share same resources for mutual benefit in this architecture. A multitenant container database (CDB) houses one or more pluggable databases. A container is defined as an aggregation of data files and metadata that exist within a CDB. The new multitenant features promotes greater replication and mobility for the databases. A pluggable database can be cloned from other databases within the CDB. A pluggable database can also easily be transferred from one CDB to another.

Each CDB contains a master set of data files and metadata known as the root container. Each CDB also contains a seed container that is used as a template for creating other PDB. Instead of having n number of separate 11g databases you can have one 12c container database that is attended to by a single set of metadata and processes and several cloned pluggable databases. (Kuhn, 671) In Oracle 12c we see that there is a division between the container and pluggable components that effectively splits the data dictionary and the background processes. The CDB owns a root namespace which includes an Oracle only part of database which allows for direct patch updates. There is also an administrative layer that allows for efficient tuning which is applied to contained pdb(s) instantaneously.

In a multitenant setup we can also as easily migrate and plug in databases as they can be created via the seed template. The pluggable databases (pdb) utilize a technique similar to transportable tablespaces to migrate tables and indexes from one template to another. These pluggable components also allow for flashback mechanisms to recover any lost data. The *create pluggable database* command creates pluggable database in the container database by copying files from the seed template. Not only can we import and create new fresh templates but we can also emulate existing pluggable databases via cloning.

Cloning existing pluggable databases can be achieved by issuing the following command. The statement *create pluggable database salespdb FROM hrpdb* allows users to clone a new pdb from an existing pdb .This type of cloning allows us to simplify many tasks related to testing and maintenance in the database environment.

Beyond cloning, we can also use the new plug in architecture to plug in an existing pdb to a separate container database. For example we can issue the statement *CREATE PLUGGABLE DATABASE salespdb USING '/disk1/usr/financepdb.xml' NOCOPY* which allows us to use a migrated pdb with our container database. We can similarly unplug a database using the third generation *plug/unplug* command of the Data Pump. (Llewellyn 48)

There are many benefits of multitenant architecture. Firstly it allows for higher consolidation density as it bypasses the schema based consolidation problems that made the database provisioning process harder in the past. With Oracle 12c, a pluggable database can be plugged into a container database which shares its memory and background processes. From this point onwards we can continue to scale up the database infrastructure. In this new architecture tasks like taking a backup of the database and preparing for disaster management is made simpler by doing these tasks at the container level. The individual pluggable components inherit the traits of the container database and are more manageable.

SQL commands allow rapid provisioning using Oracle 12c architecture. It takes just seconds to issue out commands via SQL related to cloning, plugging in and taking snapshots. Furthermore the new database also succeeds in creating new paradigms for rapid patching and upgrades. To patch a database you can unplug a pdb and patch it to the appropriate container that has the correct version of the software. To upgrade a database we can install a new copy of a database and then upgrade the pointers in the user database to a new copy of the root database. This way updating the metadata is a thousand times faster. (Jackson)

PDB based consolidation has the same SGA and background processes as schema based consolidation. Furthermore the logical virtualization within SGA, redo, and undo logs brings new capability for within CDBs and between PDBs. This allows for PDB point in time recovery. (Llewellyn, 51)

Furthermore it’s easy to migrate current Oracle 11g databases to Oracle 12c. We don’t require any application or database code changes. We can simply plug in the existing Oracle 11g database as a CDB. By adding a container as a buffer the Oracle 12c offering promotes greater security in the database layer. It also simplifies cloud based provisioning and management across multiple data centers while keeping the applications it supports intact. In the new database architecture we have a concept of tiered storage applied to the actual database. This means that we can select any permutation of pdb(s) to simplify consolidation and performance tuning.

The new architecture can be seamlessly integrated for an Oracle Real Application cluster. The container architecture also supports dynamic data masking- which promotes greater security and resilience for sensitive data. Big data discovery processes can be augmented via row pattern matching features that column and graph data models implement. (Garret, 3) The new pluggable architecture brings with it greater security, granularity, synergy, provisioning and transferability. (Kuhn, 673)

Conclusion

The benefits of this new architecture are already apparent for customers who tried the Oracle 12c beta program. Among these customers Aramark reported that utilizing the new Oracle 12c database multitenant features resulted in savings of cost and improved latency. Aramark consultants concluded that the new container based approach made testing and development easier for the organization and noted that they did not have to worry about hardware choices in the future. (Garrett, 3)

Similarly consultants from Postbank, Germany’s largest retail banks found the Oracle 12c beta version features tremendously helpful in upgrading databases. Consultants noted that downtime was shorter and the upgrade process completed in a shorter and quicker cycle. Additionally they noted that table specific recovery instead of tablespace recovery, optimized memory allocation. The 12c efficient patching capabilities were also highlighted. (Garrett, 4)

Overall, the new multitenant pluggable databases mark several improvements some of which are increased scalability, optimized resource utilization, improved performance, rapid provisioning, consolidated management and patching. This new approach brings true within database virtualization on the offering table. Over time Oracles product portfolio has only evolved to support the demands of new IT practices like infrastructure virtualization, cloud computing, mobile computing and big data technologies. Pluggable databases have no doubt made the job of the DBA easier and more efficient.

References

Coronel, Carlos & Morris, Steven. (2015). Database Systems: Design, Implementation, and Management   11thEdition. Cengage Learning.

Jackson, J. (2012). How Oracle's pluggable databases will work. Retrieved April 26, 2016, from <http://www.computerworld.com/article/2492055/database-administration/how-oracle-s-pluggable-databases-will-work.html>

Garrett, Brian, Quinn, Evan. (2013). Introducing Oracle Database 12c: Manage Many Databases as One. Washington DC: Enterprise Strategy Group Press.

Kuhn, Darl. (2013). Pro Oracle database 12c administration. New York, NY: Apress.

Llewellyn, Bryn. (2013). Oracle Multitenant. Redwood Shores, California: Oracle Corporation.

Oracle Tech Writer. (2016).Oracle Database Online Documentation 12c Release 1. Retrieved April 27, 2016, from <https://docs.oracle.com/database/121/CNCPT/cdbovrvw.htm#CNCPT89234>