Oracle FLEXCUBE Private Banking

May 2011

Database configuration parameters and Recommended Schedule Maintenance

Oracle Part Number E51529-01

Document Control

Date	Version	Description	Author
30th November 2008	1.0	Initial Design	PWM DBA Team
23rd May 2009	1.1	Updates to the document	Vijay Nair
27th JULY 2009	1.2	Added schedule maintenance recommendation	Jaimin Shah
24th Aug 2009	1.3	Added tablespace size and recommended storage parameters	Jaimin Shah
31st Oct 2009	1.4	Changes for internationalization	VJ
09th Mar 2010	1.5	Added FCPBS_TEMP tablespace size	Jaimin Shah
07 April 2010	1.6	Changed to include additional tablespaces	VJ
18 May 2011	1.7	Product Changes	VJ

Table of Contents

1 Database and File system Recommendation

1.1 Introduction

This document is intended to configure the Oracle FCPB database parameters and file system recommendation for the Production/UAT/TEST environment as per the environment.

Please note that this document covers only the parameters that are required for initial database setup and does not cover the schema setup which is done post the configuration of the database expressed in this document.

1.2 Effective Storage

The product would require an effective storage of **400 GB** achieved via an internal/external storage device. The effective storage is arrived at based on the volumes provided by the bank (including migration)

1.3 File System Recommendation

The database instance would have to be created with the following parameters

DATABASE NAME:

FCPB<<Environment>><<Number>> e.g. FCPBUAT01, FCPBPROD01

FILE SYSTEM:

The following file systems would need to be created:

- File systems for data,indexes,temporary,undo,interfaces and partitioning (Around 70% of the disk space)
- File system for backup (For this file system along with the file system for archive remaining 30% disk space should be split as per the bank’s policy)
- File system for archive

On Unix/Linux like systems on separate mount points:

```
/data/oracle/FCPBUAT01/fcpbdata
/data/oracle/FCPBUAT01/fcpbidx
/data/oracle/FCPBUAT01/fcpbint
/data/oracle/ FCPBUAT01/fcpbundo
/data/oracle/ FCPBUAT01/fcpbpart
/data/oracle/FCPBUAT01/backup
/data/oracle/FCPBUAT01/archive
```

TABLESPACES:

Table space Name	Minimum Size for Datafile	Mount Point	Recommended storage parameters
PB_USR	5 GB	fcpbdata	EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO DEFAULT TABLESPACE
PB_IDX	5 GB	fcpbidx	EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO
PB_INT	7 GB	fcpbint	EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO NOLOGGING
PB_UND	5 GB	fcpbundo	UNDO TABLE SPACE LEAVE DEFAULT VALUES
PB_RDN	200 MB	fcpbdata	EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO
PB_PRT	10 GB	fcpbpart	EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO
PB_MSC	10 GB	fcpbdata	EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO
PB_TMP	8 GB	fcpbtmp	TEMPORARY TABLE SPACE LEAVE DEFAULT VALUES

REDOLOG File Size – Recommended Redolog file size is 500 MB hold at least 20 minutes of redo information. Separate mount point is preferred for the same, but the bank can choose to implement it as per the standards

DATABASE CHARACTER SET - AL32UTF8

NATIONAL CHARACTER SET – AL16UTF16

DATABASE LOCALE - American

1.4 Database Configuration Parameters

Once the installation is complete, the following parameters have to be reflected in the instance that is deemed for use by the Oracle FCPB Application.

Parameter Name	Value
statistics_level	ALL
memory_target	7620M
memory_max_target	7620M
db_cache_size	0
java_pool_size	0
large_pool_size	0
streams_pool_size	0
shared_pool_size	0
db_keep_cache_size	160M
workarea_size_policy	AUTO
pga_aggregate_target	0
db_block_size	8K
cursor_sharing	FORCE
db_file_multiblock_read_count	128
undo_retention	1800
optimizer_index_caching	80
optimizer_index_cost_adj	20
query_rewrite_enabled	TRUE
query_rewrite_integrity	TRUSTED

1.5 Oracle Initialization Parameters

This section describes detailed discussion of the Oracle initialization parameters.

1. STATISTICS_LEVEL is a parameter that affects various types of statistics gathering and DBA advisories.

2. MEMORY_TARGET: In Oracle Database 11*g,* the Automatic Shared Memory Management (ASMM) feature is introduced to automatically determine the size of the database buffer cache (default pool), shared pool, large pool, and Java pool through use of the initialization parameter MEMORY_TARGET. The initialization parameter STATISTICS_LEVEL must also be set to TYPICAL (default) or ALL to use ASMM.

3. MEMORY_MAX_SIZE specifies the maximum size of SGA and PGA for the lifetime of the instance.

4. PGA_AGGREGATE_TARGET specifies the total amount of session PGA memory that Oracle will attempt to allocate across all sessions. PGA_AGGREGATE_TARGET was introduced in Oracle 9*i* and should be used in place of the *_SIZE parameters such as SORT_AREA_SIZE. Also, in Oracle 9*i,* the PGA_AGGREGATE_TARGET parameter does not automatically configure ALL *_SIZE parameters. For example, both the LARGE_POOL_SIZE and JAVA_POOL_SIZE parameters are not affected by PGA_AGGREGATE_TARGET. The advantage of using PGA_AGGREGATE_TARGET is the ability to cap the total user session memory to minimize OS paging. Oracle uses the PGA_AGGREGATE_TARGET to allocate memory for sorting only if the WORKSPACE_POLICY is set to AUTO. If it is not, Oracle will use the older manual method of managing sort area, which includes setting the SORT_AREA_SIZE and HASH_AREA_SIZE parameters. When PGA_AGGREGATE_TARGET is set, WORKAREA_SIZE_POLICY must be set to AUTO. In Oracle 11g there is no need not to specify any value for this parameter.

5. WORKAREA_SIZE_POLICY used to enable automatic PGA size management.

6. DB_KEEP_CACHE_SIZE is used to hold tables that are frequently referenced by the application, such as small tables that have frequent full table scans and reference tables for the application.

7. CURSOR_SHARING converts literal SQL to SQL with bind variables, reducing parse overhead. Sql statements were not shared due to the usage of literals. This resulted in additional hard parses which were consuming significant database time. To get rid of hard parses, we have to set the parameter "cursor_sharing" to "force".

8. OPTIMIZER_INDEX_COST_ADJ coarse adjustment between the cost of an index scan and the cost of a full table scan. Setting this parameter to a value would pretty much guarantee index use.

9. OPTIMIZER_INDEX_CACHING lets you adjust the behavior of cost-based optimization to favor nested loops joins and IN-list iterators.

10. QUERY_REWRITE_ENABLED Used to enable Materialized View and Function-Based-Index capabilities.

11. QUERY_REWRITE_INTEGRITY Used to enable Materialized View and Function-Based-Index capabilities.

12. DB_FILE_MULTIBLOCK_READ_COUNT For full table scans to perform I/O more efficiently, this reads this many blocks in a single I/O.

13. UNDO_MANAGEMENT and UNDO_TABLESPACE Oracle store multiple copies of data that is currently being modified by a user. While the transaction that modifies data is in progress, a copy of the original data is used to provide a read-consistent view of the database for other sessions. In addition, the modifying users may choose to undo work by issuing a ROLLBACK statement, or their process could crash in the middle of the transaction, requiring Oracle to undo their work in progress to restore the database to a consistent state.

To support each of these scenarios, Oracle stores the pre-edited data in a special data structure, an undo or rollback segment. You can set the UNDO_MANAGEMENT and UNDO_TABLESPACE parameters so Oracle will automatically create and manage undo segments.

To enable automatic undo management, first set UNDO_MANAGEMENT to auto. Next, set the UNDO_TABLESPACE to the name of the tablespace that will store the system-managed undo segments.

2 Database and File system Recommendation

2.1 Analyzing tables and indexes

Oracle FCPB database performance can be improved by analyzing tables and indexes. Analyzing tables and indexes are extremely important for regular maintenance of Oracle FCPB database. Performed regularly, this function can help keep performance of the versioned Oracle FCPB database at a maximum.

Below script that analyzes the Oracle FCPB schema.

```
begin
dbms_stats.gather_schema_stats(
ownname => 'PRODUCTIONSCHEMANAME',
estimate_percent => null,
method_opt => 'FOR ALL COLUMNS SIZE AUTO',
degree => 2,
cascade => TRUE
);end;
```

This script is not a solution for every performance related issue, but are something that should be done on a regular basis as scheduled maintenance.

Analyzing the data should be done regularly on a dataset that is changing. This updates the statistics that are used by the Oracle Cost Based Optimizer, to work out the best execution plan for SQL

2.2 Caching look up tables

Cache can be best defined as the process of storing frequently used data in the memory, usually data that is costly to retrieving, for reuse, typically this data is stored in memory since retrieving data from memory is much more efficient than retrieving the data from disk.

With the respect of oracle when a SQL statement requests a row from a table, Oracle first checks the internal memory structures to see if the data is already in a data buffer. If the requested data is there, it's returned, saving a physical IO operation. We recommend using KEEP pool to catch look up tables.

KEEP pool is used to hold tables that are frequently referenced by the application, such as small tables that have frequent full table scans and reference tables for the application.

Below mentioned script should be configure as a part of weekly scheduled maintenance

Pls refer to pbs_buffer_pool_script.sql as part of the installation pack

pbs_buffer_pool_script.sql

2.3 Shrinking segments

A shrink operation is an online in-place operation that results in the high water mark being reset after compacting the Oracle blocks making them denser and releasing unnecessary free blocks back to the tablespace. These free blocks can be used for other segments requiring additional blocks.

Shrinking can improve performance of queries and DML operations since Oracle has to read fewer blocks especially during full table scans. In some cases even problems such as row-chaining may be reduced if Oracle touches the block during the shrinking process.

Below mentioned script should be configure as a part of weekly scheduled maintenance.

Below mentioned script that shrinking segments.

Pls refer to pbs_shrink_pool_script.sql

pbs_shrink_pool_script.sql

Database configuration parameters and Recommended Schedule Maintainance.
May 2011
Version number 1.7

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com/ financial_services

Due care has been taken to make this Database configuration parameters and Recommended Schedule Maintainance Document as accurate as possible. However, Oracle Financial Services Software Limited makes no representation or warranties with respect to the contents hereof and shall not be responsible for any loss or damage caused to the user by the direct or indirect use of this document and the accompanying Software System. Furthermore, Oracle Financial Services Software Limited reserves the right to alter, modify or otherwise change in any manner the content hereof, without obligation of Oracle Financial Services Software Limited to notify any person of such revision or changes.

All company and product names are trademarks of the respective companies with which they are associated.