Oracle Database= Physical files+Oracle Instance

Oracle Instance= physical memory+ background process

Physical memory- SGA (System or Share Global Area)

PGA- private global area (PGA)

PGA is allocated to user at the time of successful connection

Every DML operations internally follows Read operation

Update, Delete always the respective data will read from disk to memory (Database buffer cache)

Commit 10 AM- CheckPoint

Update

Create

Commit 10:10AM

Log Buffer

LGWR

log seq=1 log seq=2

Log group 2

Log group1

log seq=4

log seq=3

log switch – lgwr process is switching from one group to another

When logwriter transfers the instructions from log buffer to redo log file; control file is updated with checkpoint information. at the log switch log writer will start writing on next group. Data related to first group is transferred to data file and the datafile header is update with checkpoint info.

If controlfile checkpoint info is same as datafile header checkpoint information. Database does not require recovery

Status of log file

Current- lgwr is writing on group

Inactive- lgwr is not writing and the respective contents are transferred from database buffer cache to datafiles; It can be dropped.

Active- lgwr is not writing and respective contents are not transferred from database buffer cache to datafiles; This group cannot be overwritten by lgwr

Unused- new group

Archive Mode

Some offline location

Restaurant

10 serviceman

30 customers

Request queue

Response queue

Dispatcher is a process who is going to dispatch the shared server process to any user

Init<sid>.ora –initialization parameter file – you would like to change any value of parameter; shutdown and restart the instance-Increase the unavailability

Spfile<sid>.ora- server parameter file – can change the parameter dynamically without shutting down the instance - increase the availability

Log Buffer

Dirty List

LRU List

DBWR LGWR

grid

Tablespace offline – Tablespace is not visible to the database

Tablespace ReadOnly- It is only available for reading purpose

Controlfile always checks the consistency of the database

Relogfile checkpoint information with checkpoint information in datafiles header

If redologfile checkpoint information is greater than (32- checkpoint numbers) datafile header (28-checkpoint numbers) information means recovery is required

REDO – redo and rollback the uncommitted transactions

Startup nomount – only instance is created

Startup mount – instance is created, controlfiles are opened

Startup open or startup – instance is created, control files are opened , redolog files and data files are opened

System – data dictionary, sysaux – performance statistics

Create database orclman

User sys identified by admin

User system identified by admin

Logfile

Group 1 (‘D:\oracle\oradata\ORCLMANUAL\Redolog\redo1.log’) size 100M,

Group 2 (‘D:\oracle\oradata\ORCLMANUAL\Redolog\redo2.log’) size 100M

datafile

‘D:\oracle\oradata\ORCLMANUAL\datafile\system.dbf’ size 2G autoextend on

Sysaux datafile ‘D:\oracle\oradata\ORCLMANUAL\datafile\sysaux.dbf’ size 1G autoextend on

Default temporary tablespace temp

Tempfile ‘D:\oracle\oradata\ORCLMANUAL\datafile\temp.dbf’ size 1g

Undo tablespace undotbs1

Datafile ‘D:\oracle\oradata\ORCLMANUAL\datafile\undo.dbf’ size 1G;

You create a directory

Yuu create three separate folders – control file, data file,redolog file

You create initialization parameter file

Manadatory contents in pfile

db\_block\_size=8192

db\_name="orclman"

compatible=19.0.0

control\_files=("D:\oracle\oradata\ORCLMANUAL\ControlFile01.CTL",

"D:\oracle\oradata\ORCLMANUAL\ControlFile02.CTL")

undo\_tablespace=UNDOTBS1

oradim –new –sid <sid name>- creation of service

set oracle\_sid=<new sid name>

sqlplus/nolog

conn /as sysdba

startup nomount pfile=’pfile location with name”

startup instance

create spfile from pfile

shutdown

startup nomount

SQL> Create database orclman

2 User sys identified by admin

3 User system identified by admin

4 logfile

5 group 1('d:\oracle\oradata\orclmanual\redolog\redo1.log') size 100m,

6 group 2('d:\oracle\oradata\orclmanual\redolog\redo2.log') size 100m

7 datafile

8 'D:\oracle\oradata\ORCLMANUAL\datafile\system.dbf' size 2G autoextend on

9 Sysaux datafile 'D:\oracle\oradata\ORCLMANUAL\datafile\sysaux.dbf' size 1G

autoextend on

10 Default temporary tablespace temp

11 Tempfile 'D:\oracle\oradata\ORCLMANUAL\datafile\temp.dbf' size 1g

12 Undo tablespace undotbs1

13 Datafile 'D:\oracle\oradata\ORCLMANUAL\datafile\undo.dbf' size 1G;

Database created.

@D:\WINDOWS.X64\_193000\_db\_home\rdbms\admin\catalog.sql

@D:\WINDOWS.X64\_193000\_db\_home\rdbms\admin\catproc.sql

Startup

Instance starts-> startup nomount

Control files are opened-> startup mount

Datafiles and redolog files are opened-> startup open

Startup or startup open

Shutdown

Datafiles and redolog files are closed

Database dismounted-> control file is closed

Instance shuts down

Shutdown abort

Datafiles and redolog files are not closed i.e. these files are going to be in inconsistent state

Database dismounted-> control file are not closed- these files are going to be in inconsistent state

Only instance shuts down

Database startup requires instance recovery done by SMON process

Rollforward

Database will open

Rollback

Pfile – initialization parameter file

Init<sid>.ora

Static file- changes can take effect on next startup of the instance

In increases the unavailability of the database for new parameter to take effect

Spfile<sid>.ora

Dynamic file –changes can take effect on nextstartup or for current instance, or for both the time.

Scope=spfile <change for next startup of instance>

Scope= memory<change for currently running instance>

Scope=both<for the current instance and next startup of instance>

Default location

Oraclehome\database

Spfile<sid>.ora-highest priority

Spfile.ora

Init<sid>.ora(pfile)

Init.ora->lowest priority

Multiplexing the controlfile using spfile

Show parameters control\_files

Alter system set control\_files=’first\_file’,’second file’,’newfile name’ scope=spfile

Shutdown

Copy the database any control file to new location and rename it.

Startup

Show parameters control\_files

Query will start at 10:00 AM 101 to 1110 11:00AM EMP- Transaction Started

Transaction will begin at 10:10 AM

10:30AM

Small transactions

Commit or rollback

Snapshot too old error

Undo\_retention

Undo tablespace

Old Data of Table which is getting updated by transaction

Tablespace-> Segments->Extents->Blocks

At the time of table creation a segment is allocated

Data dictionary was supposed to be updated

Dictionary Managed Tablespace (DMT)

Locally Managed Tablespace (LMT) – Space management is done in Tablespace or in datafile header itself before reaching to data dictionary. Hence the contention on data dictionary is reduced

In dictionary

DBA\_ -> Database wide views

ALL\_-> views which are owned by You and for which you have access

USER\_-> views which are owned by you

DATAbase

Automatic Storage Management (Network Attached Storage) (Logical Volume Manager)

Redundancy

RBAL

Oracle19C grid Architecture

Schema – logical name given to set of objects

Database is a schema-> users

User is a schema->Tables

Table is a schema->datatypes

Char –i(3) string length is always 3

‘ab ‘ – performance wise char datatype is good; space is wasted

Varchar2- variable length string (3)

2 -strlenght

1create user testuser identified by admin default tablespace bigtbs

2quota unlimited on bigtbs;

grant create session, create table to testuser;

conn testuser/admin

QL> select constraint\_name,Table\_name from user\_constraints

2 where table\_name like 'TESTTBL%';

Drop – Remove from the data dictionary -DDL

Truncate – wipe of all the data and release the storage space - DDL

Delete- deletion by row and do not release the space-DML

Primary and unique key always creates the unique index

Delete remove the data from table and corresponding index entry

Cardinality is high for primary key columns due to unique rows

Btree Index -OLTP

Car table-2000

Color - Blue, Red,white,green – cardinality is low

OLAP

Blue 1011001…2000

Red 0 100110…2000

With sequence two pseudo columns exists

Currval – current value of a sequence

Nextval – next value of a sequence

On commit delete rows- delete the rows at the end of transaction

Segment is made up of extents

2M 4M

Varchar2(60)

40

Row is chained - it is occupying more than one block

Row is migrated leaving pointer to previous block

Pctfree-amount of free space reserved for future updates;

Block is out of free list- no more insert can be done

The old block can be considered for insert if space utilization falls below 40%

PCT USED

We need to have a new block for new insert

HWM –High water Mark-shows the maximum space utilization for a table; which is going be reduced with truncate but not delete

SQL> SELECT tp.endian\_format

2 FROM v$transportable\_platform tp,

3 v$database d

4 WHERE tp.platform\_name = d.platform\_name;

SQL> create tablespace test\_transport

2 datafile 'D:\oracle\oradata\ORCLMANUAL\DataFile\testtrans.dbf' size 1G auto

extend on;

Tablespace created.

SQL> create user testt identified by testt

2 default tablespace test\_transport

3 temporary tablespace temp

4 quota unlimited on test\_transport;

User created.

SQL> grant create session, create table to testt;

Grant succeeded.

SQL> conn testt/testt

Connected.

SQL> create table testttbl(id number, tdesc varchar2(10));

Table created.

SQL> insert into testttble values (1,'abc');

insert into testttble values (1,'abc')

\*

ERROR at line 1:

ORA-00942: table or view does not exist

SQL> insert into testttbl values (1,'abc');

1 row created.

SQL> insert into testttbl values (2,'a1b1c1');

1 row created.

SQL> insert into testttbl values (3,'a2b2c2');

1 row created.

SQL> commit;

Commit complete.

SQL> desc dbms\_tts.transport\_set\_check(ts\_list=>'TEST\_TRANSPORT',incl\_constraint

s => TRUE);;

SP2-0565: Illegal identifier.

SQL> desc dbms\_tts.transport\_set\_check(ts\_list=>'TEST\_TRANSPORT',incl\_constraint

s => TRUE);

SP2-0565: Illegal identifier.

SQL> exec dbms\_tts.transport\_set\_check(ts\_list=>'TEST\_TRANSPORT',incl\_constraint

s => TRUE);

PL/SQL procedure successfully completed.

SQL> desc transport\_set\_violations;

Name Null? Type

----------------------------------------- -------- ----------------------------

VIOLATIONS VARCHAR2(2000)

SQL> select \* from transport\_set\_violations;

no rows selected

SQL>

SQL> alter tablespace test\_transport read only;

SQL> create or replace directory tdir as 'd:/ttbsdir';

Directory created.

SQL> grant read,write on directory tdir to system;

Grant succeeded.

C:\Users\JAY GANESH> expdp userid=system/admin directory=tdir transport\_tablespa

ces=test\_transport dumpfile=tdump.dmp logfile=tlog.log

on source database you put the exported tablespace in offline

alter tablespace <tblspace name> offline.

destination database import statement

C:\Windows\system32>impdp userid=system/admin directory=tdir dumpfile=tdump.dmp

logfile=tlog.log transport\_datafiles='D:\oracle\oradata\ORCLMANUAL\DataFile\tes

ttrans.dbf'

sysdba and sysoper

startup,shutdown, create database, drop database, spfile creation, archivelog

sysoper

Profile creation

Failed login attempts

Password life time

Paasword lock time

Cpu\_per\_session

Private\_sga

SQLPLUS

create directory expdir as 'D:\oracle\Newexpdir';

grant read,write on directory expdir to testuser;

create table sqltbl (id number, testdesc varchar2(15));

SQL Loader Control File

load data

infile 'D:\oracle\Newexpdir\sample.dat'

into table sqltbl

fields terminated by ','

(id, testdesc)

SQL loader data file

001,'ABC'

002,'A1B1C1'

003,'A2B2C2'

For remap \_schema

C:\Windows\system32>expdp system schemas=testuser directory=expdir dumpfile=dump

dir.dmp

C:\Windows\system32>impdp system/admin directory=expdir dumpfile=testuser.dmp re

map\_schema=testuser:testtt

remap\_tablespace

C:\Windows\system32>expdp system tablespaces=bigtbs directory=expdir dumpfile=te

stuser.dmp

C:\Windows\system32>impdp system directory=expdir dumpfile=testuser.dmp remap\_ta

blespace=bigtbs:testnew

external table creation

SQL> create table extsql (id number, testdesc varchar2(10)) organization externa

l (type oracle\_loader

2 default directory expdir

3 access parameters

4 (records delimited by newline) location (‘sample.dat'))

;

BEGIN

DBMS\_SCHEDULER.CREATE\_PROGRAM( program\_name => 'CALCSAL',

program\_action => 'TESTUSER.CALSAL’,

program\_type => 'STORED\_PROCEDURE',

enabled => TRUE);

END;

BEGIN

DBMS\_SCHEDULER.CREATE\_SCHEDULE( schedule\_name => 'stats\_schedule',

start\_date => SYSTIMESTAMP,

end\_date => SYSTIMESTAMP + 30,

repeat\_interval => 'FREQ=HOURLY;INTERVAL=1', comments => 'Every hour');

END;

BEGIN

DBMS\_SCHEDULER.CREATE\_JOB (

job\_name => ‘calsal',

job\_type => 'STORED\_PROCEDURE',

start\_date => '28-APR-08 07.00.00 PM Australia/Sydney',

repeat\_interval => 'FREQ=DAILY;INTERVAL=2',

auto\_drop => FALSE,

comments => 'My new job');

END;

Independent job

BEGIN

DBMS\_SCHEDULER.CREATE\_JOB (

job\_name => ‘TESTUSER.CALPROC10',

job\_type => 'STORED\_PROCEDURE',

job\_action => 'OPS.SALES\_PKG.UPDATE\_SALES\_SUMMARY',

start\_date => ’11-APR-2022’,

repeat\_interval => 'FREQ=DAILY;INTERVAL=2', /\* every other day \*/

end\_date => ‘12-APR-2022',

auto\_drop => FALSE,

job\_class => ‘DEFAULT\_JOB\_CLASS’,

comments => 'My new job');

END;

/

Scheduler

Program

SELECT job\_name, status, error#, run\_duration FROM USER\_SCHEDULER\_JOB\_RUN\_DETAILS where job\_name=’ TESTUSER.CALPROC10’;

BEGIN

DBMS\_SCHEDULER.**DROP\_JOB** (‘TESTUSER.CALPROC10, sys.DEFAULT\_JOB\_CLASS’);

END;

With program

BEGIN

DBMS\_SCHEDULER.CREATE\_JOB (

job\_name => ‘Job1’,

program\_name => ‘CALCSAL’,

repeat\_interval => 'FREQ=DAILY;BYHOUR=12',

comments => 'Daily at noon');

END;

With schedule

BEGIN

DBMS\_SCHEDULER.CREATE\_JOB (

job\_name => ‘job2’,

job\_type => 'PLSQL\_BLOCK',

job\_action => 'BEGIN SALES\_PKG.UPDATE\_SALES\_SUMMARY; END;',

schedule\_name => ‘stats\_schedule’);

END;

BEGIN

DBMS\_SCHEDULER.RUN\_JOB(

JOB\_NAME => ' JOB2',

USE\_CURRENT\_SESSION => FALSE);

END;

Step be step for creating a job and execution

Program

SQL> BEGIN

2 DBMS\_SCHEDULER.CREATE\_PROGRAM( program\_name => 'CALCSAL',

3 program\_action => 'TESTUSER.CALSAL (is a procedure)’,

4 program\_type => 'STORED\_PROCEDURE',

5 enabled => TRUE);

6 END;

7 /

Job

1 BEGIN

2 DBMS\_SCHEDULER.CREATE\_JOB (

3 job\_name => 'Job2',

4 program\_name => 'CALCSAL',

5 repeat\_interval => 'FREQ=DAILY;BYHOUR=12',

6 comments => 'Daily at noon');

7\* END;

QL> /

Execute the job

SQL> BEGIN

2 DBMS\_SCHEDULER.RUN\_JOB(

3 JOB\_NAME => ' JOB2',

4 USE\_CURRENT\_SESSION => FALSE);

5 END;

6 /

SQL> SELECT job\_name, status, error#, run\_duration

2 FROM USER\_SCHEDULER\_JOB\_RUN\_DETAILS where job\_name='JOB2';

JOB\_NAME