Oracle Database Online Documentation 11g Release 2 (11.2)/Database Concepts/Physical Storage Structures/Overview of Control Files

Oracle Database Online Documentation 11g Release 2 (11.2) /Database Reference/Control file

Oracle Database Online Documentation 11g Release 2 (11.2) /Database Administrator's Guide/11 Managing Control Files

10gR2 New Feature: Eliminate Control File Re-Creation (Doc ID 329981.1)

How to Recreate a Controlfile (Doc ID 735106.1)

控制文件是数据库的大脑，而 SYSTEM 表空间是数据库的心脏。本文从整体上概览控制文件并对控制文件进行初步的探讨。以能够更好的理解数据库的初始化。

>控制文件的使用

既然控制文件在数据库中扮演着重要的角色，那么控制文件中到底存储了哪些重要信息，在数据库运行过程中又是如何发挥重要作用的呢？

首先从文档上得知控制文件中保存着下列信息：

•数据库的名字 和数据库唯一标识 (DBID)

•数据库创建的时间

•数据文件、在线日志文件、归档日志文件的信息

•表空间的信息

•RMAN 备份信息

及 checkpoints信息等

既然控制文件保留了这么多信息，那么其作用就是能够初始化数据库。

1 正常打开数据库需要的文件信息：数据库文件，在线日志文件

控制文件会追踪你对数据库结构的更改，这写更改都会被数据库自动更新到控制文件中。

2 无法打开数据库需要的恢复信息：检查点

检查点指定了实例恢复的起始点，因为检查点之前所有提交的更改都被确保写入到了磁盘。

>多路复用控制文件

既然控制文件这么重要，oracle建议多路复用控制文件在不同的磁盘。多路复用在同一磁盘没有任何作用还增加io。

>控制文件结构

控制文件包括两种类型的记录：这些记录都在控制文件块中，控制文件的块大小为16k

•可循环重用的记录

这些记录都是些非关键信息，可以被覆盖重写。当控制文件中可用的记录槽用满后，数据库会自动选择扩展控制文件或覆盖重写已有的记录。例如，RMAN备份信息

•不可循环重用的记录

这部分记录包括关键信息并且不能被覆盖重写。例如表空间，数据文件，日志文件信息，这些记录数据库结构的信息，绝不能丢失，对数据库结构的更改会自动记录到控制文件中。

那么控制文件有多大，大小有限制吗，这些信息会记录多长时间呢？

答案是有大小限制，但是oracle并没有明确指出这个值。控制文件的大小，是我们间接决定的

1 控制文件大小

STATUS    NAME                            IS\_ BLOCK\_SIZE FILE\_SIZE\_BLKS

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

    +DATA/orcl/controlfile/current.260.867775171        NO     16384          594

当前控制文件的大小16384\*594

2 不可重用记录空间大小

alter database backup controlfile to trace;

or

Oracle Database Online Documentation 11g Release 2 (11.2) /Database Administrator's Guide/11 Managing Control Files/The CREATE CONTROLFILE Statement

从创建控制文件的语句中可以看到不可循环重用记录的大小

    MAXLOGFILES 16

    MAXLOGMEMBERS 3

    MAXDATAFILES 100

    MAXINSTANCES 8

    MAXLOGHISTORY 292

例如：MAXDATAFILES 100 oracle会预留100个记录槽空间用来存放数据文件

MAXLOGHISTORY 292 指定归档日志最多记录数，默认值依据cpu，内存配置设置

注意：小文件数据库，数据文件最多：65533

####################################

注意：db\_files在10gR2之后不受控制文件MAXDATAFILES限制，MAXDATAFILES会依据db\_files设置自动扩展。

同理MAXLOGHISTORY 是数据库用来决定为要记录的归档日志名分配多少空间的参数。

但是这个参数值会随着需要而自动增长，其最大值只受控制文件最大size的限制。

所以理论上讲控制文件中可记录归档日志的个数不受限制。

而控制文件最大值参考

[https://docs.oracle.com/cd/E11882\_01/server.112/e40402/limits002.htm#REFRN0042](https://docs.oracle.com/cd/E11882_01/server.112/e40402/limits002.htm" \l "REFRN0042)

如果只用控制文件保存Rman的备份记录，那么信息的保存时间主要取决于参数CONTROL\_FILE\_RECORD\_KEEP\_TIME， 这个定义值会保留能够恢复这个时间数据的所有备份信息。如果一个新纪录需要使用控制文件的可回收区，那么不受参数限制的记录就会被覆盖。所以您会发现同样的命令，但是有的可以全删，有的只删除部分，删除部分的说明记录超过了CONTROL\_FILE\_RECORD\_KEEP\_TIME中定义时间的恢复需要，并且被新纪录覆盖。您可以查询视图gv$backup\_piece 和 gv$backup\_set获取控制文件的保留的备份记录信息或者用Rman命令list backup summary; 查询。

参考文档 Relation between RMAN retention period and control\_file\_record\_keep\_time ( Doc ID 397269.1 )

对于归档日志，它的最小记录保留时间也是受CONTROL\_FILE\_RECORD\_KEEP\_TIME的限制，但是没有最大时间限制，只有最大数量限制，就是MAXLOGHISTORY参数作用, 它会间接影响归档日志在控制文件的数量，但是只要控制文件没到最大值，参数MAXLOGHISTORY可以持续自动增长，归档文件也可以继续添加。

具体请参考文档:

Why Are Controlfile Entries For Archived Logs Not Overwritten After CONTROL\_FILE\_RECORD\_KEEP\_TIME? ( Doc ID 1056085.1 )

Database SQL Language Reference - MAXLOGHISTORY Clause

Link: [http://docs.oracle.com/cd/E11882\_01/server.112/e41084/statements\_5003.htm#BEGIN](http://docs.oracle.com/cd/E11882_01/server.112/e41084/statements_5003.htm" \l "BEGIN)

您好，

感谢您的更新，当有新的归档日志需要加入控制文件时，首先，如果记录还有空间可以装，就直接加入，如果没有即视图V$CONTROLFILE\_RECORD\_SECTION 中RECORDS\_USED等于RECORDS\_TOTAL，那就查看是否旧的归档日志受到保护限制，如果没有就覆盖旧的的记录，如果不可以覆盖，控制文件就会扩展。您可以先查一下视图V$CONTROLFILE\_RECORD\_SECTION，RECORDS\_USED是否已经等于RECORDS\_TOTAL还是比它小。

参考文档：

10gR2 New Feature: Eliminate Control File Re-Creation ( Doc ID 329981.1 )

Why Are Controlfile Entries For Archived Logs Not Overwritten After CONTROL\_FILE\_RECORD\_KEEP\_TIME? ( Doc ID 1056085.1 )

V$CONTROLFILE\_RECORD\_SECTION-http://docs.oracle.com/cd/E11882\_01/server.112/e40402/dynviews\_1088.htm#BEGIN

#############################################################

10gR2之前达到了最大值需要重建控制文件吗

10gR2 New Feature: Eliminate Control File Re-Creation (Doc ID 329981.1)

3 可重用空间记录大小

NAME                     TYPE     VALUE

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

control\_file\_record\_keep\_time         integer     7

control\_file\_record\_keep\_time 指定控制文件保留可重用记录信息的最少天数，参数越大则保留信息越多，控制文件越大。

> 控制文件信息的查询

控制文件中记录的信息，可以通过动态性能视图查询。

v$database

v$datafile

v$controlfile

v$logfile

......

如:

SQL> select dbid,name,created,current\_scn from v$database;

      DBID NAME         CREATED   CURRENT\_SCN

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

1395555907 ORCL         31-DEC-14     2432686

> 控制文件和数据库初始化（controlfile and initialization）

数据库在打开时，除了进程常规的校验外还会进行一些特殊校验，比如将数据字典和控制文件进行交叉检查。无论哪种校验都会用到控制文件。

以下是已知的几种校验：

1.在数据库open的过程中,Oracle要进行检查中包含以下两个过程：

第一次检查数据文件头中的Checkpoint cnt是否与对应控制文件中的Checkpoint cnt一致.如果相等,进行第二次检查.

第二次检查数据文件头的开始SCN和对应控制文件中的结束SCN是否一致，如果结束SCN等于开始SCN,则不需要对那个文件进行恢复.

对每个数据文件都完成检查后,打开数据库.同时将每个数据文件的结束SCN设置为无穷大.

注意：详细的对比过程会在数据库初始化一节中介绍

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。

> 重建控制文件

控制文件这么重要，一但丢失有怎么办呢？

1 使用备份的控制文件还原restore

2 没有备份的话使用backup controlfile to trace script脚本重建

生成控制文件的脚本

SQL> alter session set tracefile\_identifier='bak\_control';

SQL> alter database backup controlfile to trace;

3 以上都没有需要手动编写创建的脚本

you must create a script from the beginning.

CREATE CONTROLFILE REUSE DATABASE "DBNAME" NORESETLOGS ARCHIVELOG

Follow the format listing:

- Location of redo logs.

- Location of datafiles

- Specifying the characterset.

Once you have listed all files correctly you are ready to recreate your controlfile

参考

Oracle Database Online Documentation 11g Release 2 (11.2) /Database Administrator's Guide/11 Managing Control Files

**How to Recreate a Controlfile (Doc ID 735106.1)**

###########################

  运行重建控制文件的脚本；这时候我们会有问题，控制文件的检查点SCN丢掉了，如何恢复回来的呢 ?  
     SQL> select group#, first\_change# , status from v$log ;  
     SQL> select checkpoint\_change#  from v$database ;    
     我们可以发现控制文件中记录的数据库系统检查点SCN从current的redo log中起始检查点SCN获得。  
  
     SQL> select file#,  checkpoint\_change#  from  v$datafile ;  
     SQL> select file#,  checkpoint\_change#  from  v$datafile\_header ;  
     我们可以发现控制文件中记录的数据文件检查点SCN从数据文件头部中的检查点SCN获得。  
 G.  SQL> recover database;    
 H.  SQL> alter database open ; 恢复完成后直接开启，不需要resetlogs;  
 I.  因为重建的控制文件中不包含temp空间的信息，所以还需要为临时表空间加入文件。

    SQL> alter tablespace temp add tempfile '/u01/temp01.dbf' ;

###############################################

> 控制文件探索

当然以上只是一个粗略的介绍，由于控制文件是个二进制文件，无法直接打开查看，但是通oracle提供的命令可以将控制文件内容转储出来便于查看：

以下是来自 Oracle Database 11g 的转储测试：  
 SQL> alter session set events 'immediate trace name controlf level 8';

SQL> select value from v$diag\_info where name='Default Trace File';  
 VALUE  
 --------------------------------------------------------------------------------  
 /opt/oracle/diag/rdbms/11gtest/11gtest/trace/11gtest\_ora\_5910.trc  
 注意：从 11g 开始，可以通过 v$diag\_info 获得当前会话转储文件的名称。

###################################################

dump and backup controfile

1. 在主库和备库分别 dump 控制文件，并上传生产的trace文件：

alter session set events 'immediate trace name controlf level 8';  
 oradebug setmypid  
 oradebug tracefile\_name;

select \* from v$diag\_info;

2 备份控制文件-重建脚本

SQL> alter session set tracefile\_identifier='bak\_control';

SQL> alter database backup controlfile to trace;

########################################################

打开这个跟踪文件现在就可以清晰的看到控制文件的内容，最开始的一段是关于数据库ID、名称等的概要信息：

DUMP OF CONTROL FILES, Seq # 7619 = 0x1dc3

\*\*\* 2016-06-09 21:26:55.178

 V10 STYLE FILE HEADER:

    Compatibility Vsn = 186646528=0xb200000

**Db ID=1395555907=0x532e7e43, Db Name='ORCL'**

    Activation ID=0=0x0

    Control Seq=7619=0x1dc3, File size=594=0x252

    File Number=0, Blksiz=16384, File Type=1 CONTROL

接下来是数据库条目的详细信息，包括了数据库的名称、数据文件及日志文件的数量、

数据库的检查点及 SCN 信息等：

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DATABASE ENTRY

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

 (size = 316, compat size = 316, section max = 1, section in-use = 1,

  last-recid= 0, old-recno = 0, last-recno = 0)

 (extent = 1, blkno = 1, numrecs = 1)

 12/31/2014 16:39:31

 DB Name "ORCL"

 Database flags = 0x40404000 0x00001000

 Controlfile Creation Timestamp  12/31/2014 16:39:34

 Incmplt recovery scn: 0x0000.00000000

 Resetlogs scn: 0x0000.000f30dc Resetlogs Timestamp  12/31/2014 16:39:34

 Prior resetlogs scn: 0x0000.00000001 Prior resetlogs Timestamp  09/17/2011 09:46:04

 Redo Version: compatible=0xb200000

 #Data files = 6, #Online files = 6

**Database checkpoint: Thread=1 scn: 0x0000.0024b514**

**Threads: #Enabled=1, #Open=1, Head=1, Tail=1**

 enabled  threads:  01000000 00000000 00000000 00000000 00000000 00000000

  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

再接下来是检查点记录信息,这部分内容包含了 Low Cache RBA 和 On Disk RBA 信息,在  
 执行数据库实例恢复时,前者是恢复的起点,后者是恢复的终点,其分别指向了日志文件中的确  
 定地址:  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 CHECKPOINT PROGRESS RECORDS   
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 (size = 8180, compat size = 8180, section max = 11, section in-use = 0,  
   last-recid= 0, old-recno = 0, last-recno = 0)  
 (extent = 1, blkno = 2, numrecs = 11)  
 THREAD #1 - status:0x2 flags:0x0 dirty:688  
low cache rba:(0x1b.16c04.0) on disk rba:(0x1c.a1c.0)   
 on disk scn: 0x0000.0008ed61 07/07/2008 08:01:59  
 resetlogs scn: 0x0000.00000001 07/04/2008 13:51:34

**heartbeat: 659440589 mount id: 1478276654**

先引用到这里，之后的学习中应该仔细阅读接下来的每个条目。

在上面的引用中， 已经频繁出现了 checkpoint 和 SCN 信息， Oracle 数据库在内部通过 SCN和检查点来保证数据库的一致性、可恢复性等重要属性，接下来会介绍 Oracle的 SCN 与 检查点机制。

关于日志历史纪录numrecs对应MAXLOGHISTORY 292

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

LOG FILE HISTORY RECORDS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

 (size = 56, compat size = 56, section max = 292, section in-use = 68,

  last-recid= 68, old-recno = 1, last-recno = 68)

 (extent = 1, blkno = 95, **numrecs = 292)**

Earliest record:

 RECID #1 Recno 1 Record timestamp  12/31/14 16:42:24 Thread=1 Seq#=1 Link-Recid=0 kccic-Recid=2

  Low scn: 0x0000.000f30dc 12/31/14 16:39:34 Next scn: 0x0000.000fcc66

Latest record:

 RECID #68 Recno 68 Record timestamp  06/09/16 12:00:42 Thread=1 Seq#=68 Link-Recid=67 kccic-Recid=2

  Low scn: 0x0000.00242b9a 06/09/16 00:16:59 Next scn: 0x0000.0024b514

 RECID #67 Recno 67 Record timestamp  06/09/16 00:16:59 Thread=1 Seq#=67 Link-Recid=66 kccic-Recid=2

<<orcl\_ora\_29689\_bak\_control.trc>>

文档原创，转载请注明出处-------------------