

关于Oracle数据库热备份脚本深入剖析 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/143/2021_2022__E5_85_B3_E4_BA_8EOrac_c102_143005.htm

我们都知道Oracle的备份有几种方式，冷备，热备，rman，imp等，我们注意到当我们采取热备的时候，需要对每个要备份的表空间置为backup模式。

通常的热备脚本都是这样的: `alter tablespace XXX begin backup.cp XXXalter tablespace XXX end backup.`（这里需要注意一点，Oracle的最小存储单位是一个数据块，一个块的大小通常设置为8KB，而操作系统的块通常是512B，这样的话一个Oracle的数据由很多个操作系统的块组成。而且对于一个数据文件来说，它的所有块对应的操作系统的块并不是按顺序存储的，当运行cp等操作系统命令时并不能指定从那个Oracle数据块开始拷贝。）当open数据库的时候，Oracle会去比较控制文件中数据文件记录和数据文件头的checkpoint cnt，如果两者相同，则判断不需要介质恢复，如果不同，这时候Oracle就会报某某文件需要介质恢复。然后拷贝回数据文件备份我们开始recover，这时候就从上次做备份时的scn开始恢复，运用日志，直到恢复结束。当cp数据文件时，比如说我们拷贝的第一个块可能是scn为100的数据块，当我们完成这个块的拷贝后，这个块有可能被别的进程多次修改，scn变为900。我们知道当数据库发生检查点时会去更新数据文件头和控制文件中的checkpoint scn，如果当我们在cp数据文件的同时发生了n次checkpoint，这时候数据文件头的scn可能被更新了很多次。这时候cp的进程去拷贝数据文件头所在的操作系统的块，可能这个数据文件头的块因为被checkpoint了很多次导致它

的scn为1000，这时候整个数据文件会出现不一致，当用这个备份文件去恢复时，恢复进程会从scn = 1000开始恢复，这样的话开始那个scn=100的块将丢失从scn100-sc1000的数据，因为数据块并不应用scn在1000以前的日志,而且这样做的话可能出现一些数据块的corruption，所以不置成backup模式备份的话并不可取。当然，如果你能确保当cp的时候不发生checkpoint，或者你的操作系统块的大小不小于Oracle的数据块大小，这些情况下不置backup mode拷贝出来的文件也是有效的。现在我们知道了为什么不能不设置backup模式，下面来讲讲alter tablespace XXX begin backup做了什么？当数据文件置于backup模式时，Oracle会去锁定数据文件头，这时候数据库发生检查点的话将不会修改文件头的checkpoint scn，而只是增加checkpoint cnt，所以不管执行cp的时候操作系统块的拷贝顺序是如何，Oracle总会从文件头的scn开始恢复，这样的话也就避免了数据丢失和数据块corruption。如果大家用的是rman来备份，那么就不会有这个问题，因为rman备份的时候rman会去对比数据块的头尾标志，如果发现不一致，那么它将会再去读这个块，直到读到一致的块才往备份集里写。但是alter tablespace XXX begin backup带来的另一个问题是会导致产生多余的日志，通过一个小小的试验就可以证明这一点。

```
SQL> 0select name,value from v$sysstat where name=redo size.NAME
VALUE-----
-----redo size 43408SQL> 0update test set a=a.1 row
0updated.SQL> commit.Commit complete.SQL> 0select
name,value from v$sysstat where name=redo size.NAME
```

VALUE-----
redo size 44060 SQL> ALTER SYSTEM DUMP LOGFILE
/netappredo/redo05.log.System altered.一个0update的动作产生
44060 - 43408 = 652bytes的redo,把表空间置为backup mode:
SQL> alter tablespace test begin backup.Tablespace altered.SQL>
0select name,value from v\$sysstat where name=redo size.NAME
VALUE-----

----redo size 44732SQL> 0update test set a=a.1 row 0updated.SQL>
commit.Commit complete.SQL> 0select name,value from v\$sysstat
where name=redo size.NAME

VALUE-----

-----redo size 53560SQL> alter tablespace test end
backup.Tablespace altered.一个0update的动作产生53560 - 44732
= 8828bytes的redo,看看到底是记了些什么? SQL> ALTER
SYSTEM DUMP LOGFILE /netappredo/redo05.log.System
altered.REDO RECORD - Thread:2 RBA: 0x00004e.000000b0.0128
LEN: 0x01b0 VLD: 0x01SCN: 0x0000.19ed24f7 SUBSCN: 1
06/29/2004 15:05:32CHANGE #1 TYP:0 CLS:29 AFN:33
DBA:0x08400029 SCN:0x0000.19ed24f2 SEQ: 1 OP:5.2..... (改动向
量1, 记载对undo header事务表的修改)CHANGE #2 TYP:0
CLS:30 AFN:33 DBA:0x0840002e SCN:0x0000.19ed24f0 SEQ: 1
OP:5.1..... (改动向量2, 记载对undo block的修改)CHANGE #3
TYP:2 CLS: 1 AFN:51 DBA:0x0cc0000f SCN:0x0000.19ed24e8
SEQ: 1 OP:11.5KTB Redo (改动向量3, 记载对数据块的修改,
也就是在数据块上执行0update test set a=a) op: 0x11 ver: 0x01
op: F xid: 0x0007.001.00014ece uba: 0x0840002e.0859.38Block

cleanout record, scn: 0x0000.19ed24f7 ver: 0x01 opt: 0x02, entries follow... itli: 1 flg: 2 scn: 0x0000.19ed24e8KDO Op code: URP row dependencies Disabled xtype: XA bdba: 0x0cc0000f hdba: 0x0cc0000bitli: 2 ispac: 0 maxfr: 4858tabn: 0 slot: 0(0x0) flag: 0x2c lock: 2 ckix: 0ncol: 1 nnew: 1 size: 0col 0: [2] c1 02CHANGE #4 MEDIA RECOVERY MARKER SCN:0x0000.00000000 SEQ: 0 OP:5.20.....(改动向量4 , 一些标记) 100Test 下载频道开通 , 各类考试题目直接下载。详细请访问 www.100test.com