MySQL 8.0 Redo Re-designed VS Oracle Redo

@zhenxu.li(Roger)

个人介绍

李真旭　云和恩墨服务交付副总经理兼开源架构部负责人

网名Roger，近10 年的Oracle技术积累
ACOUG核心会员
2014年被授予 Oracle ACE 称号
致力于技术分享与传播
ACOUG和数据库大会演讲者

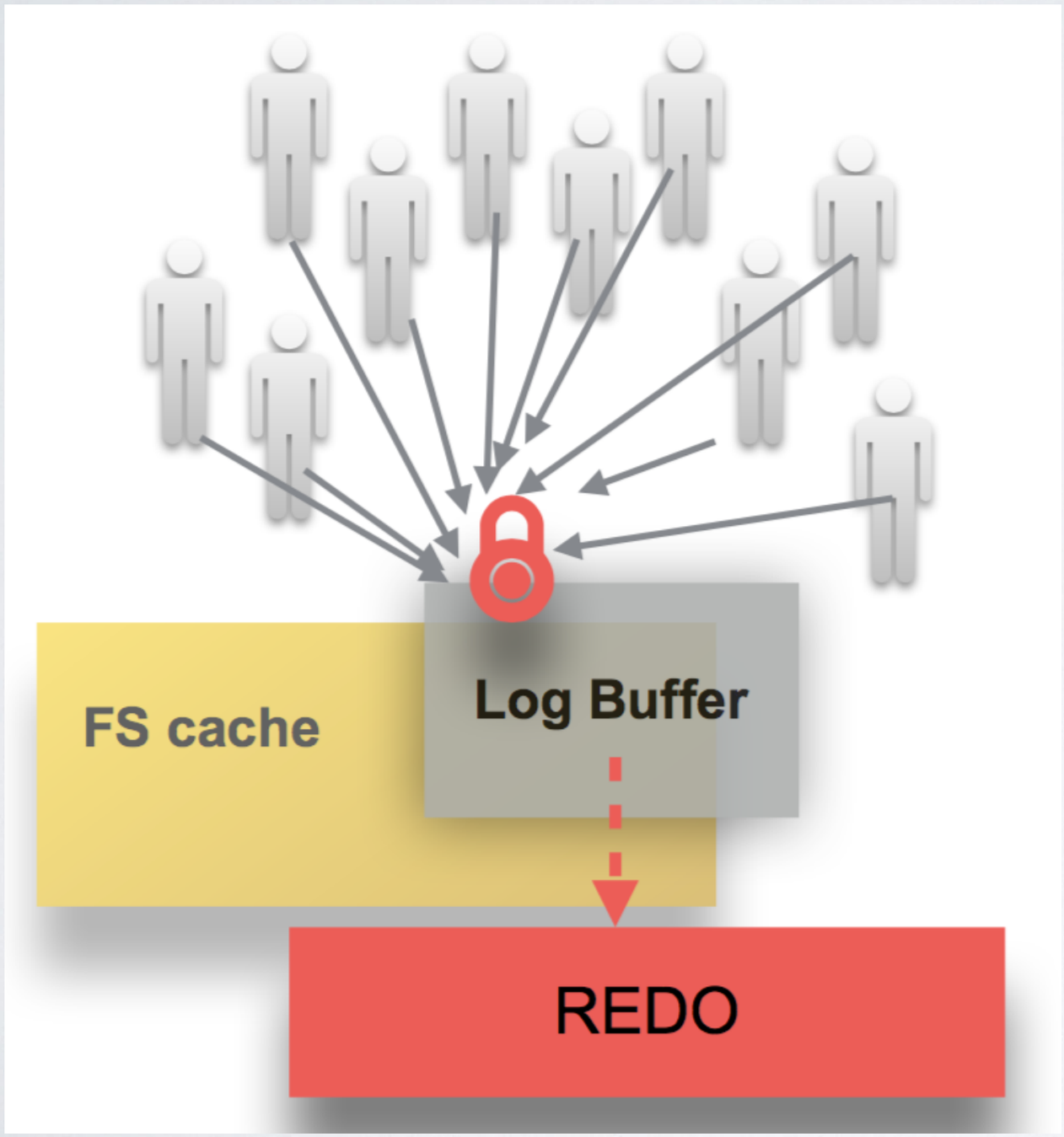
云和恩墨 国内综合数据服务领导者

汇聚 Oracle ACE专家，SQL大赛冠军，以及数十位OCM专家，同时具备MySQL和DB2专家；
为包括电信、金融、保险、电商、能源等行业500多家客户提供服务和解决方案；

About Redo

- ACID
- Redo Logging
- 事务恢复

How to flush log



最大的性能瓶颈
log_sys->mutex

行业的做法

1. InnoSQL

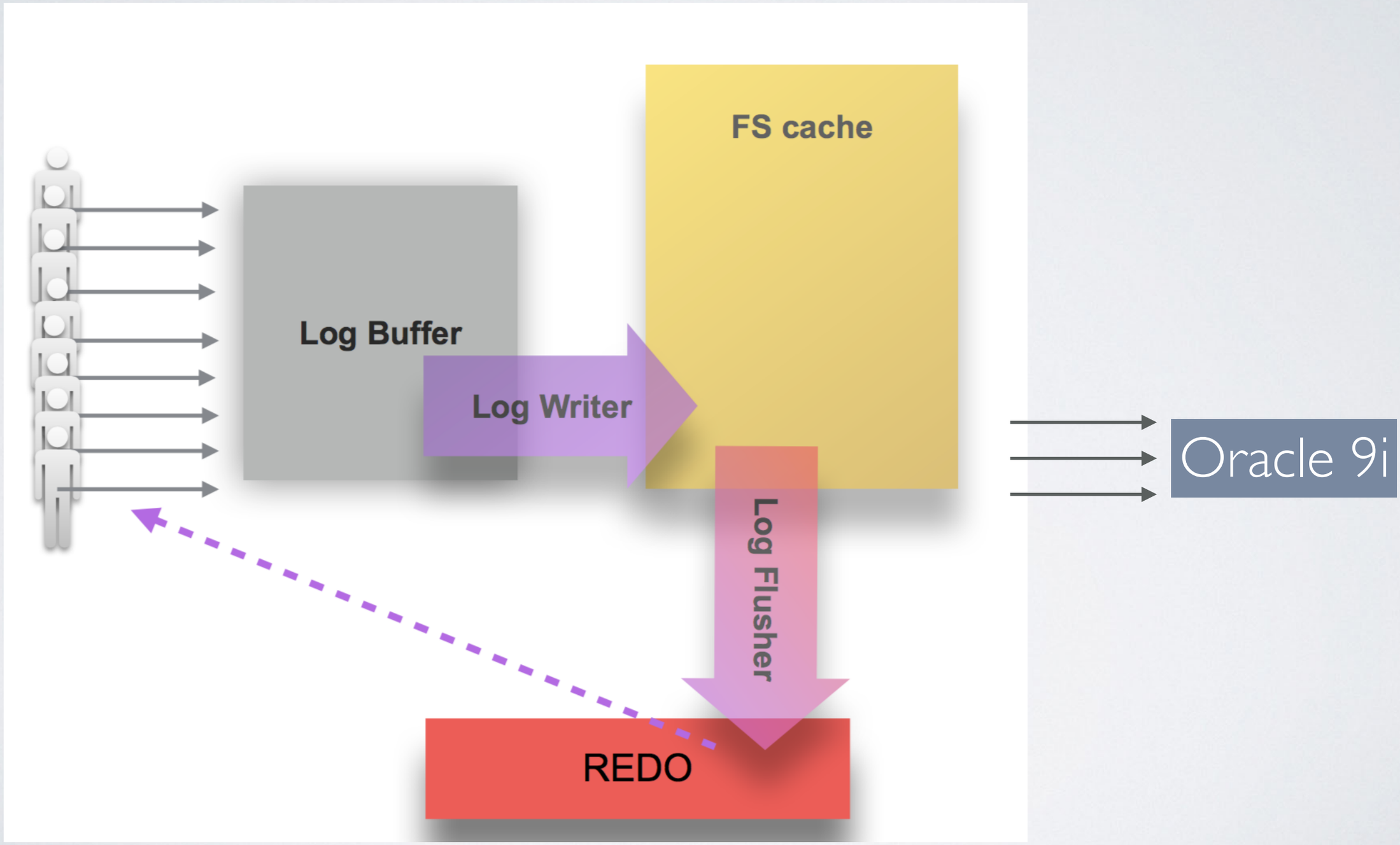
借鉴了Oracle的思路：在redo buffeer中完成空间分配即释放Mutex

2. TXSQL

引入双缓冲区机制 & w_mutex 锁，flush redo log的过程中释放 log_sys->mutex，继续持有 log_sys->w_mutex，从而阻塞写，不阻塞LSN相关的读操作，flush完成后释放w_mutex。

说明：此图来自腾讯TXSQL的分享

MySQL 8.0—How to do



Oracle log buffer Flush的实现

1) PGA中产生Redo Entries
2) Server Process获取Redo Copy latch(CPU_COUNT*2)
3) Server Process获取redo allocation latch(仅1个)
4) 在log buffer中分配空间
5) 释放redo allocation latch
6) 将Redo Entry写入Log Buffer
7) 释放Redo Copy latch

Oracle 9i之前

Oracle 9.2

9.2 引入Log Buffer Parallel机制

```
SQL> alter system set log_parallelism=4 scope=spfile;

System altered.
.......
SQL> select LATCH#,name,level# from v$latch_children where name='redo allocation';

    LATCH# NAME                                                  LEVEL#
---------- ---------------------------------------------------------- ----------
       115 redo allocation                                             5
       115 redo allocation                                             5
       115 redo allocation                                             5
       115 redo allocation                                             5
```

Shared Strand

Oracle 10.2 Redo Latch

```
wwww.killdb.com@select * from V$sgastat where name like '%private strand%';

POOL        NAME                         BYTES
----------- -------------------------- ----------
shared pool  private strands              1198080

www.killdb.com@select trunc(value * KSPPSTVL / 100) * 65 * 1024
  2  from (select value from v$parameter where name = 'transactions') a,
  3       (select val.KSPPSTVL
  4          from sys.x$ksppi nam, sys.x$ksppsv val
  5         where nam.indx = val.indx
  6           AND nam.ksppinm = '_log_private_parallelism_mul') b;

TRUNC(VALUE*KSPPSTVL/100)*65*1024
---------------------------------
                          1198080
```

Oracle 10g引入private strands结构

```
www.killdb.com@select LATCH#,name,level#,gets,misses from v$latch_children where name='redo allocation';

    LATCH# NAME                                                  LEVEL#       GETS     MISSES
---------- ---------------------------------------------------------- ---------- ---------- ----------
       150 redo allocation                                             5        335          0
       150 redo allocation                                             5         16          0
       150 redo allocation                                             5        258          0
       150 redo allocation                                             5         14          0
       150 redo allocation                                             5          2          0
       150 redo allocation                                             5          2          0
       150 redo allocation                                             5          2          0
       150 redo allocation                                             5          2          0
       150 redo allocation                                             5          2          0
       150 redo allocation                                             5          2          0
       150 redo allocation                                             5          2          0
       150 redo allocation                                             5          2          0
       150 redo allocation                                             5          2          0
       150 redo allocation                                             5          2          0
       150 redo allocation                                             5          2          0
       150 redo allocation                                             5          2          0
       150 redo allocation                                             5          2          0
       150 redo allocation                                             5          2          0
       150 redo allocation                                             5          2          0
       150 redo allocation                                             5          2          0
```

哪些是Private？哪些是Public？

为什么要引入private strands？有什么意义？

Oracle 11.2 Redo Latch

```
SQL> select name,misses,count(1) from v$latch_children where name like '%redo%'
  2  group by name,misses order by 1,2;

NAME                                 MISSES   COUNT(1)
------------------------------ ---------- ----------
redo allocation                        2513          1
redo allocation                        8352          1
redo allocation                      520157          1
redo allocation                     9709105          1
redo copy                                 0        126
redo copy                                 1          2

6 rows selected.
```

```
SQL> show parameter cpu_count

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------
cpu_count                            integer     64
```

为什么Redo allocation latch越来越少?

```
QL> set lin 200
QL> select latch#,name,count(1) from v$latch_children
 2  where name like '%redo%' group
 by latch#,name;
 3
   LATCH# NAME                                                      COUNT(1)
--------- ---------------------------------------------------------- ----------
      209 redo allocation                                                   169
      208 redo copy                                                          12

L>
```

Oracle 12.2 Redo Latch

```
SQL> select con_id,latch#,name,count(1) from v$latch_children
  2  where name like '%redo%'
  3  group by
  4  con_id,latch#,name
  5  order by 1,3;

    CON_ID     LATCH# NAME                           COUNT(1)
---------- ---------- ------------------------------ ----------
         0        349 redo allocation                        52
         0        348 redo copy                              2
```

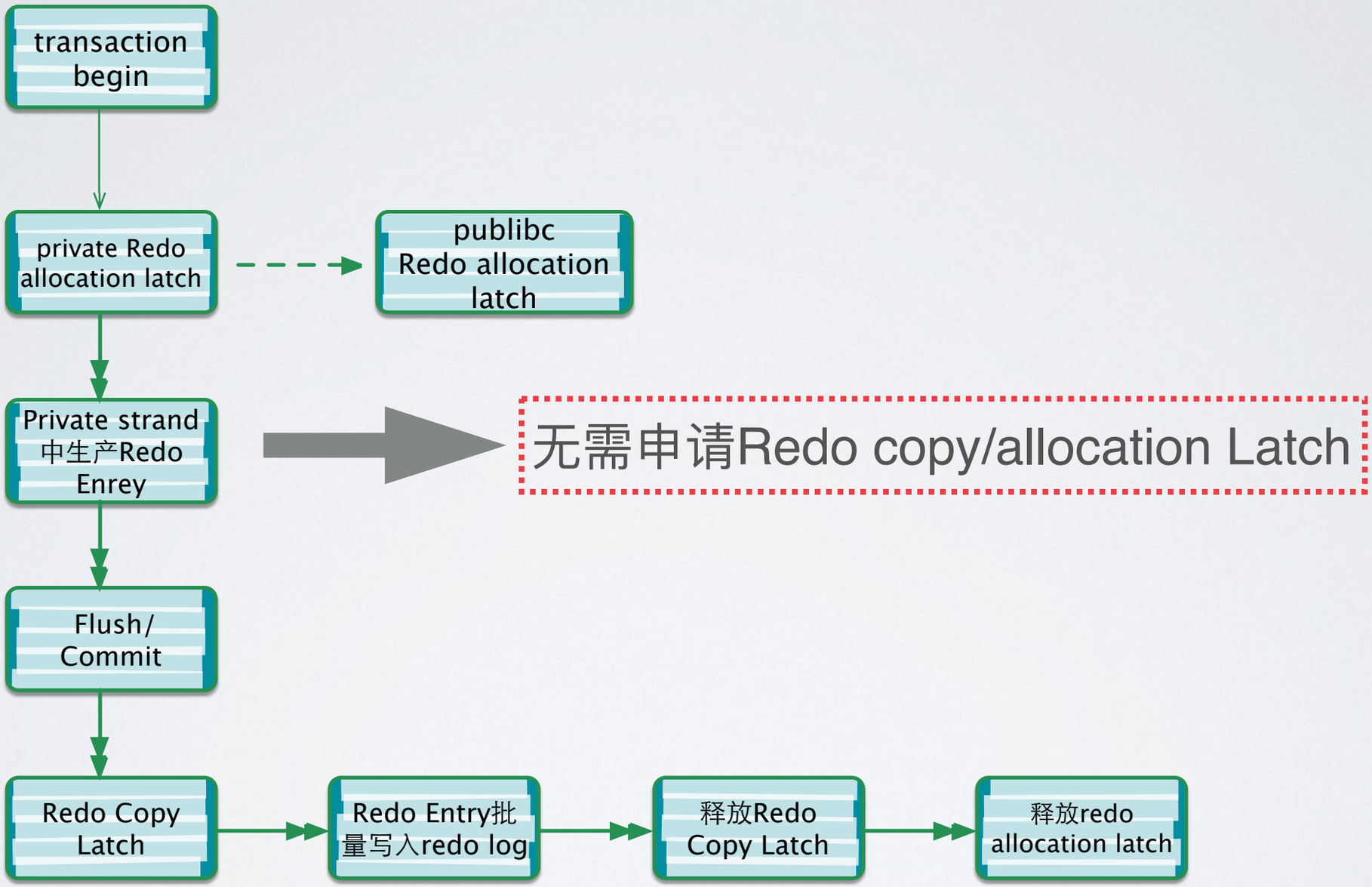
```
SQL> select con_id,strand_size_kcrfa,count(1) from x$kcrfstrand
  2 where last_buf_kcrfa = '00'
  3 group by con_id,strand_size_kcrfa;

    CON_ID STRAND_SIZE_KCRFA   COUNT(1)
---------- ----------------- ----------
         0            132096         51
```

Oracle 18c

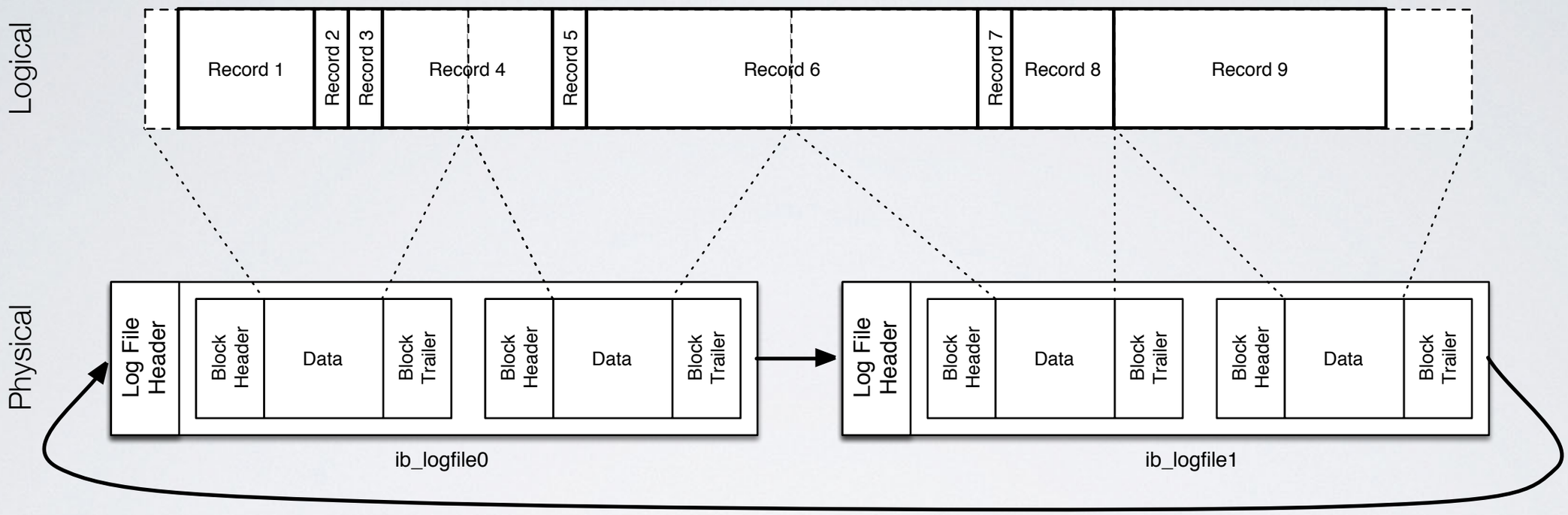
```
SQL> select * from V$sgastat where name like '%private strand%';

POOL         NAME                            BYTES     CON_ID
------------ -------------------------- ---------- ----------
shared pool    private strands                6945792         1

SQL> select strand_size_kcrfa,count(1) from  x$kcrfstrand
  2   where last_buf_kcrfa = '00'
  3  group by strand_size_kcrfa;

STRAND_SIZE_KCRFA   COUNT(1)
----------------- ----------
           132096         51

SQL> select 51*(129+4)*1024 from dual;

51*(129+4)*1024
---------------
        6945792
```

Oracle 18c看上去并没有改变

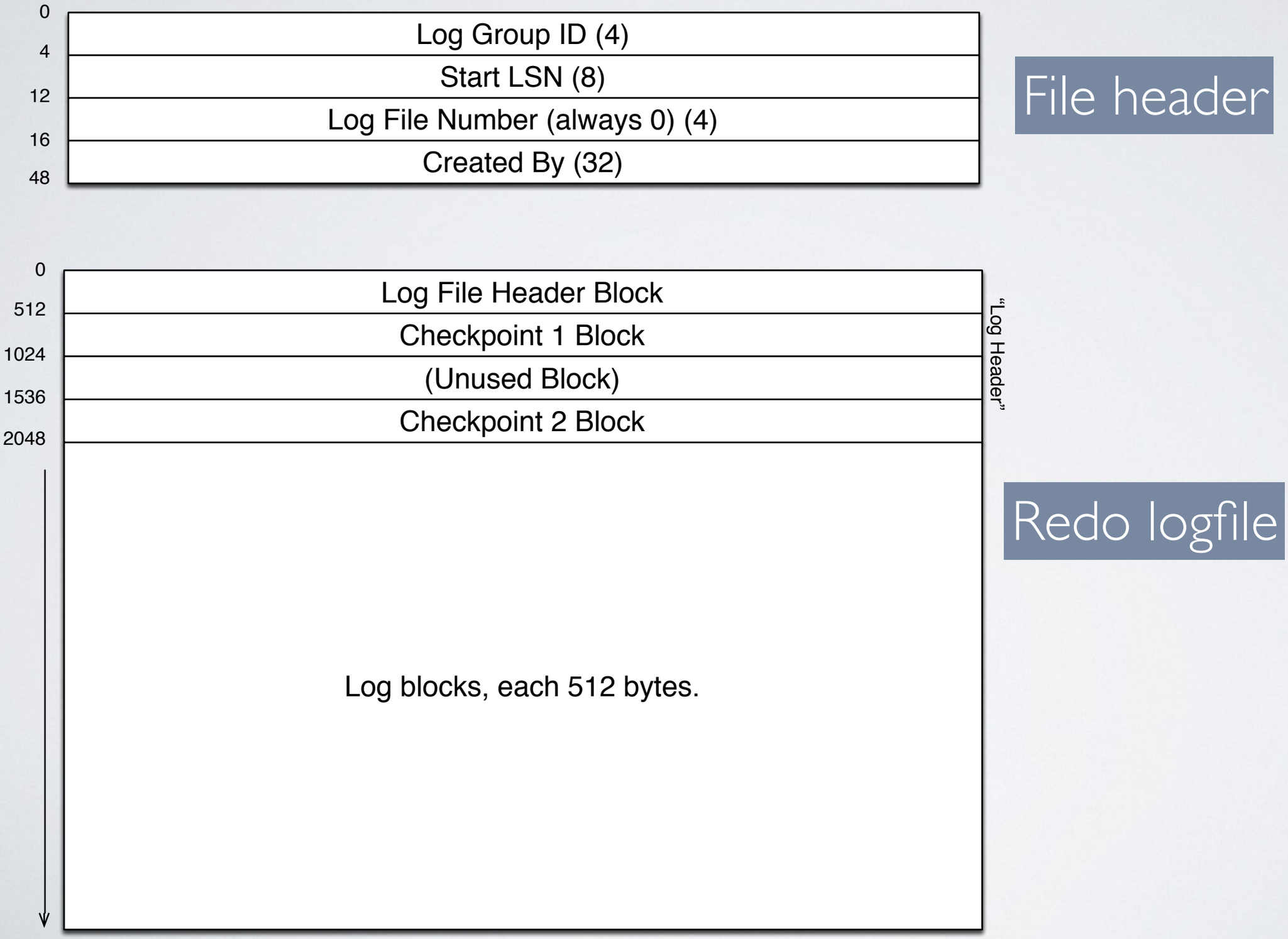
新版本的流程是怎么样的

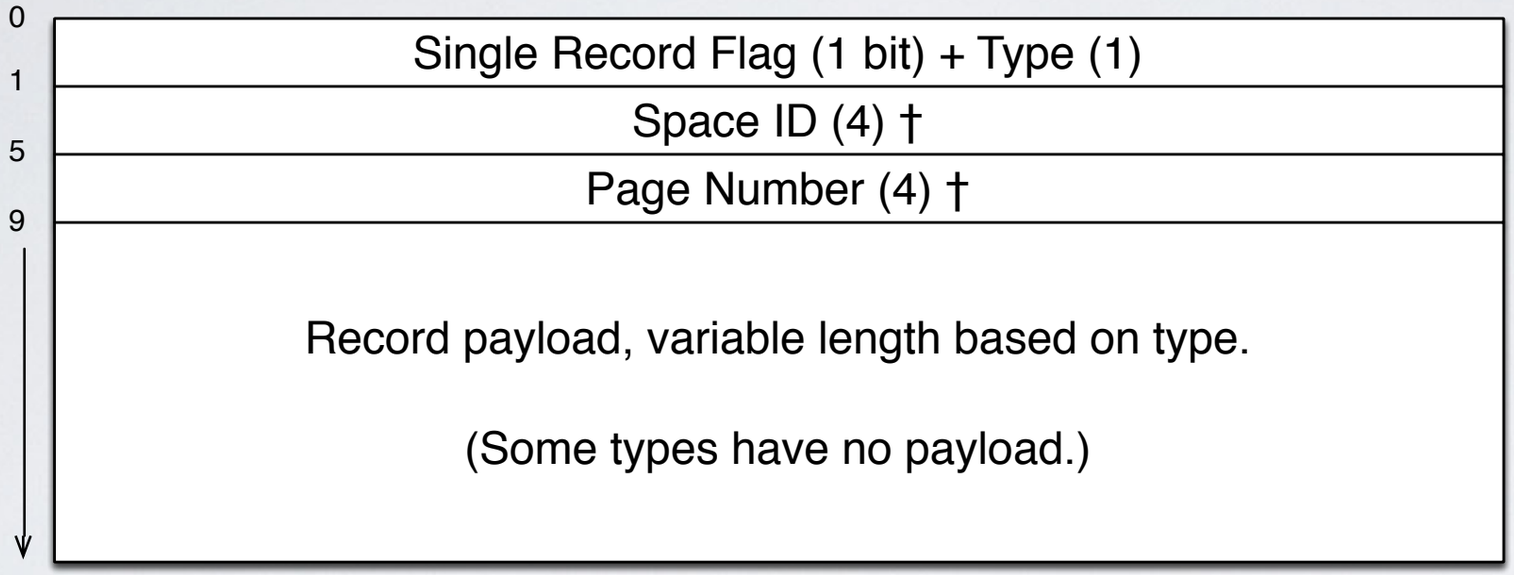


Redo log结构的差异性-MySQL



Redo log结构的差异性-MySQL





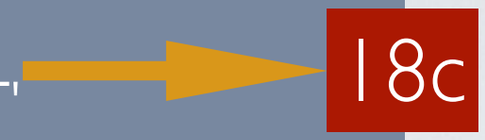
log record

{Space ID + Page NO+操作类型+数据}

Redo log结构的差异性-Oracle

Oracle 18c redo log dump

FILE HEADER:
Compatibility Vsn = 301989888=0x12000000
Db ID=2273241221=0x877ee885, Db Name='TEST'
Activation ID=2273216645=0x877e8885
Control Seq=1375=0x55f, File size=409600=0x64000
File Number=1, Blksiz=512, File Type=2 LOG
descrip:"T 0001, S 0000000004, SCN 0x000000000017316f-0xffffffffffffffff"
thread: 1 nab: 0xffffffff seq: 0x00000004 hws: 0x1 eot: 1 dis: 0
resetlogs count: 0x3a1ec407 scn: 0x0000000000016ac11
prev resetlogs count: 0x39aaecfc scn: 0x0000000000000001
Low scn: 0x000000000017316f 05/02/2018 21:21:42
Next scn: 0xffffffffffffffff 01/01/1988 00:00:00



Low scn: 0x0e72.b387becf (15885801078479) 05/04/2018 14:38:57
Next scn: 0xffff.ffffffff 01/01/1988 00:00:00
.....
redo log key is 266573fd4b2bd927e0ea601e958e12b1
redo log key flag is 5
Enabled redo threads: 1 2 3 4

Old: 6 Byte=48 bit Max_SCN=2^{48}
New: 8 Byte=64 bit Max_SCN=2^{64}

Most recent redo scn: 0x0000000000000000
Largest LWN: 0 blocks
End-of-redo stream : No
Unprotected mode
Miscellaneous flags: 0x800000
Miscellaneous second flags: 0x0
Thread internal enable indicator: thr: 0, seq: 0 scn: 0x0000000000000000
Zero blocks: 0
Format ID is 18
redo log key is 2f1092d7bc36e7124fd23eaa4915f42c
redo log key flag is 5
High watermark block number: 0
Enabled redo threads: 1

BTW: About Big SCN

```
BBED> p kcvfhckp
BBED> p kcvfhhdr
 struct kcvfhhdr, 76 bytes                @20
    ub4 kccfhswv                          @20       0x00000000
    ub4 kccfhcvn                          @24       0x12000000
    ub4 kccfhdbi                          @28       0x877ee885
    ......
BBED> p kcvfhckp
struct kcvfhckp, 36 bytes                 @484
    struct kcvcpscn, 8 bytes              @484
       ub4 kscnbas                        @484      0x001aa506
       ub2 kscnwrp                        @488      0x8000
       ub2 kscnwrp2                       @490      0x0000
    ub4 kcvcptim                          @492      0x3a2098ea
    ub2 kcvcpthr                          @496      0x0001
    .......
    ub1 kcvcpetb[7]                       @519      0x00
```

无论是Data Block还是Index block都做了类似改变

Oracle 18c redo record dump

```
REDO RECORD - Thread:1 RBA: 0x000004.00000002.0010 LEN: 0x008c VLD: 0x05 CON_UID: 1
SCN: 0x0000000000173176 SUBSCN:  1 05/02/2018 21:22:02
(LWN RBA: 0x000004.00000002.0010 LEN: 0x00000001 NST: 0x0001 SCN: 0x0000000000173175)
CHANGE #1 MEDIA RECOVERY MARKER CON_ID:1 SCN:0x0000000000000000 SEQ:0 OP:24.4 ENC:0 FLG:0x0000
......
REDO RECORD - Thread:1 RBA: 0x000004.00000004.0010 LEN: 0x01d4 VLD: 0x05 CON_UID: 1
SCN: 0x000000000017317b SUBSCN:  1 05/02/2018 21:22:03
(LWN RBA: 0x000004.00000004.0010 LEN: 0x00000002 NST: 0x0001 SCN: 0x000000000017317b)
CHANGE #1 CON_ID:1 TYP:0 CLS:25 AFN:4 DBA:0x010000c0 OBJ:4294967295 SCN:0x000000000017311a SEQ:1 OP:5.2
ENC:0 RBL:0 FLG:0x0000
ktudh redo: slt: 0x0001 sqn: 0x00000359 flg: 0x0452 siz: 160 fbi: 0
            uba: 0x01001bac.00d3.16    pxid:  0x0000.000.00000000        pdbid:1
CHANGE #2 CON_ID:1 TYP:0 CLS:26 AFN:4 DBA:0x01001bac OBJ:4294967295 SCN:0x0000000000173119 SEQ:5 OP:5.1
ENC:0 RBL:0 FLG:0x0000
ktudb redo: siz: 160 spc: 5004 flg: 0x0012 seq: 0x00d3 rec: 0x16
            xid:  0x0005.001.00000359
......
```

1) 如何降低Redo日志产生量 ？
2) How to tuning redo event ?

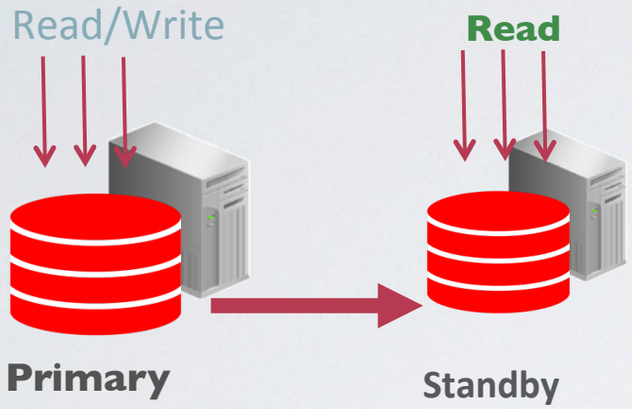
Oracle 18c — Automatic Databas

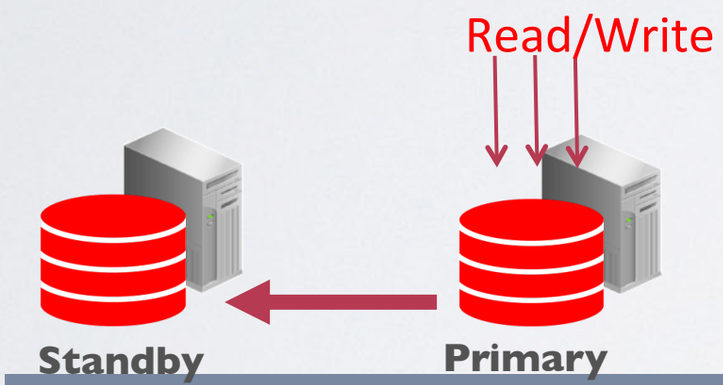
```
_use_adaptive_log_file_sync                          FALSE
_adaptive_log_file_sync_use_polling_threshold        200
_adaptive_log_file_sync_use_postwait_threshold       50
_adaptive_log_file_sync_sched_delay_window           60
_adaptive_log_file_sync_poll_aggressiveness          0
_adaptive_log_file_sync_high_switch_freq_threshold   3
```

Bug 22832181 : ADAPTIVE LOG FILE SYNC CAUSES PERFORMANCE DOWN

```
_adaptive_log_file_sync_use_postwait_threshold_aging   1001
_adaptive_log_file_sync_sampling_count                 128
_adaptive_log_file_sync_sampling_time                  3
```

Oracle 18c — Automatic Databas

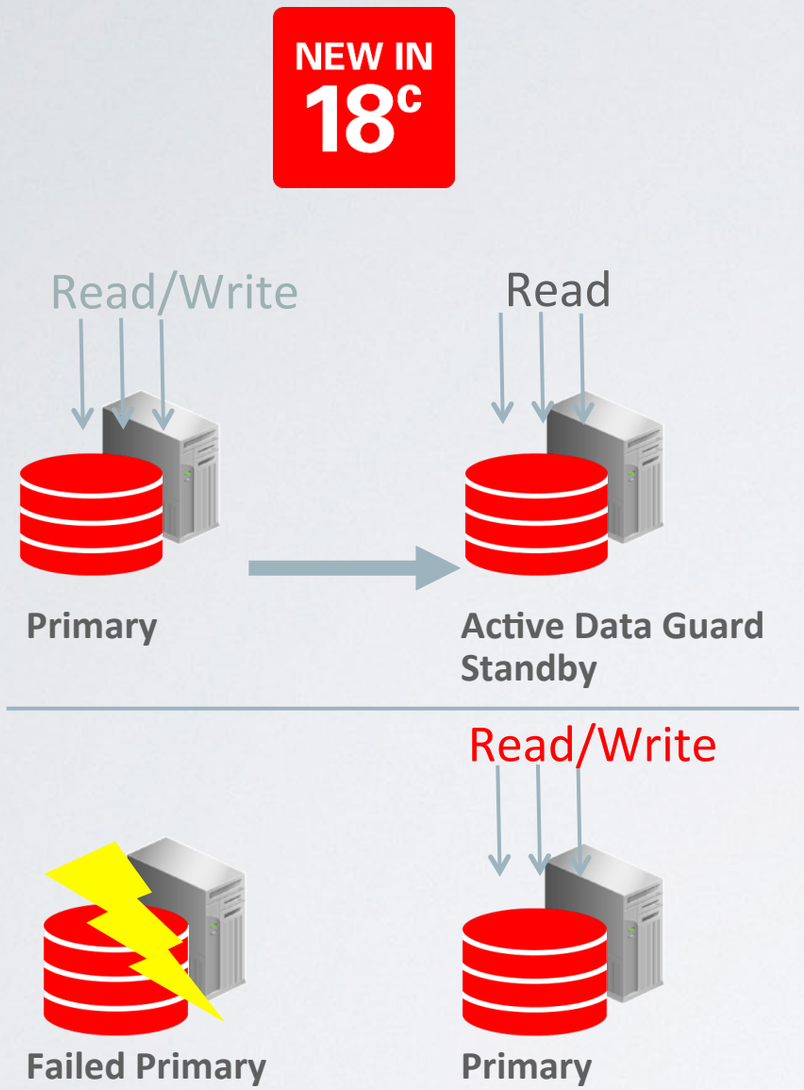




Oracle 12.2

ADG 上的只读会话在切换期间
Read-only sessions connected to Active Data Guard will
在Failover / Switchover 过程中保持连接
Remain connected during the failover/switchover
当 ADG 成为主库后由只读切换为读写模式
Become read/write after Active Data Guard becomes the primary
极大缩减了切换后的连接风暴
Dramatically reduces brownout due to reconnect storm after failover
Another reason to use Active Data Guard (instead of Data Guard)

Oracle 18c — Automatic Database



1）The database buffer cache state will be maintained on an ADG standby during a role change

2）Automatic, nothing to set up.

Services need to be configured correctly

Q&A