中亦科技
China Etek Service

Perfecting IT service and favoring clients 'success
锻造凝练IT服务 助推用户事业发展

Oracle故障与优化案例分享

小y(黄远邦)@中亦科技
13701026113 & 51994106@qq.com
微信： shadow-huang-bj

关于小y-技术即人生

中亦科技
China Etek Service

黄远邦

联系方式:
13701026113
Shadow.huang@
ce-service.com.cn
51994106@qq.com

微信号：
Shadow-huang-bj

微信号：Shadow-huang-bj

就职于北京中亦安图科技股份有限公司
中亦科技是国内领先的第三方综合运维服务提供商,提供包括数据库、操作系统、中间件、存储、网络等全方位的运维服务

十年以上Oracle数据库维护经验，擅长数据库架构设计、复杂故障、复杂性能问题定位和解决。

数年来为数十家银行总行客户提供数据库专家服务
此外为航空、证券、基金、保险、运营商、政府、制造业等众多客户提供数据库维护服务

案例-Latch竞争

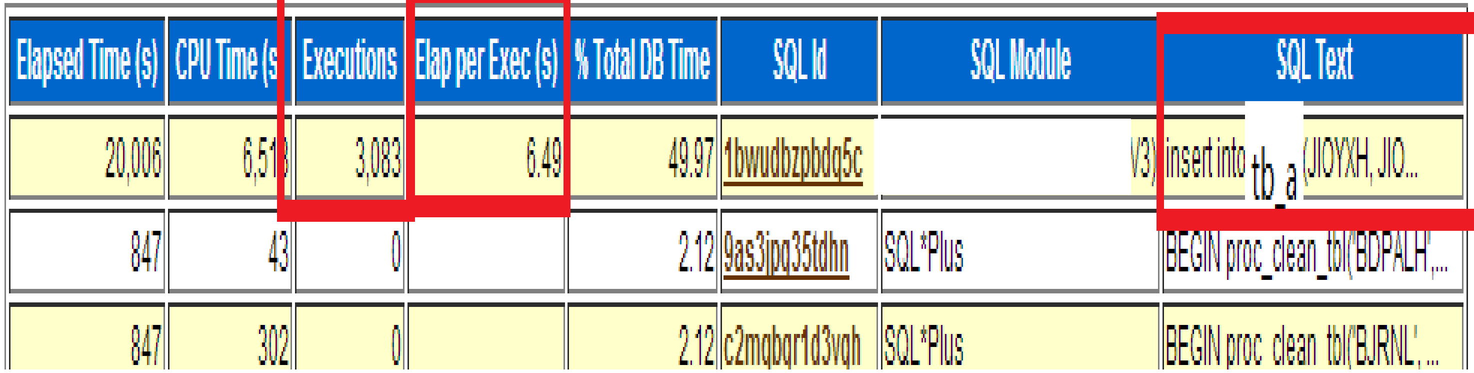
Event	Waits	Time(s)	Avg Wait(ms)	% Total Call Time	Wait Class
CPU time		63,441		26.1	
latch: undo global data	19,378,032	52,996	3	21.8	Other
db file sequential read	9,253,102	44,043	5	18.1	User I/O
db file scattered read	2,049,763	14,120	7	5.8	User I/O
latch: cache buffers chains	6,162,598	9,537	2	3.9	Concurre

- 该等待事件的解释：
- This latch serializes the access to the Undo (aka Rollback) segment information
- in the SGA. Every time a session wants to know about the state of the Undo Segments, it has to get this latch.

- They protect a linked list and hash buckets much in the same way as the hash buckets and
- buffer cache hash chains latch does. The buckets are used for quick access to undo segment
- information to reduce row cache undo$ reads.
- Each bucket points to a list of undo segment structures (kturt).

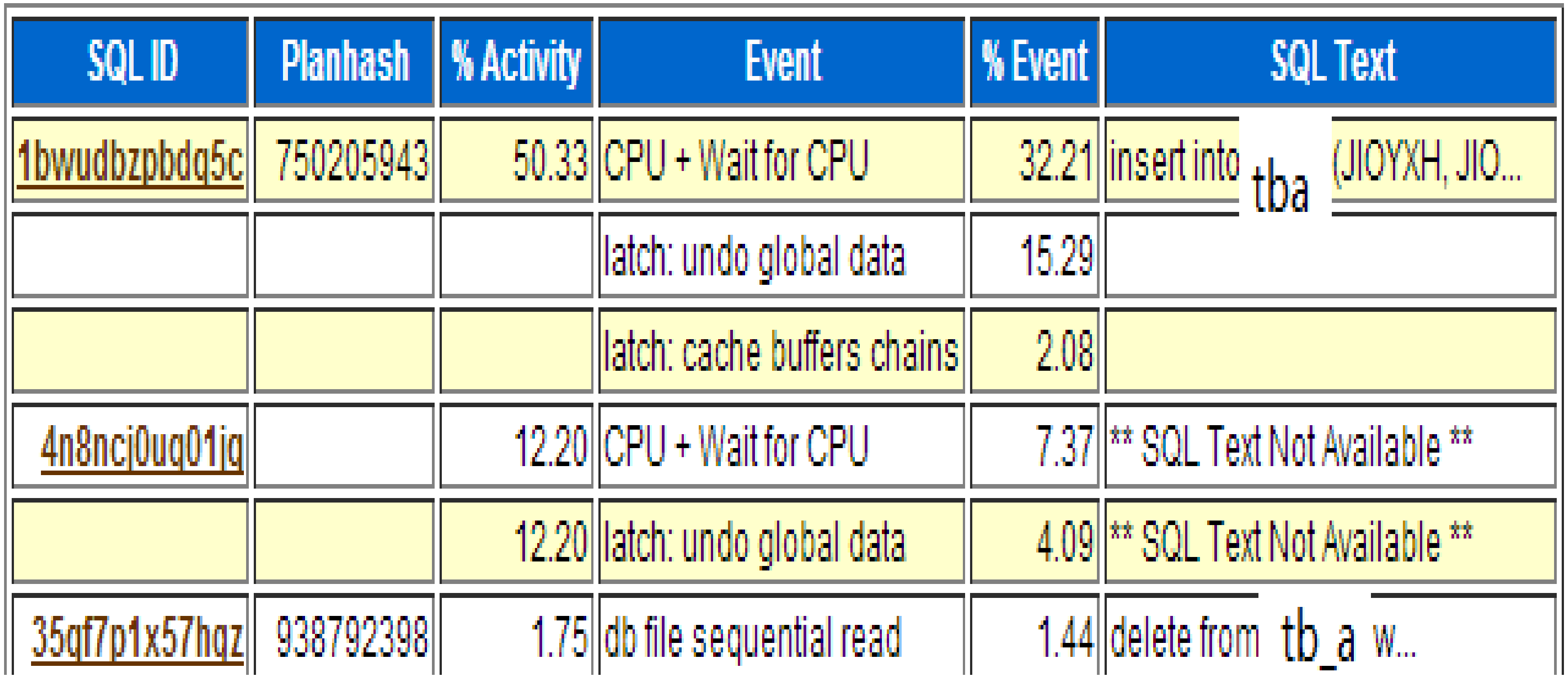
- 该函数的解释
- ktudba - Kernel Transaction Undo convert from usn to DBA

SQL ordered by Elapsed Time

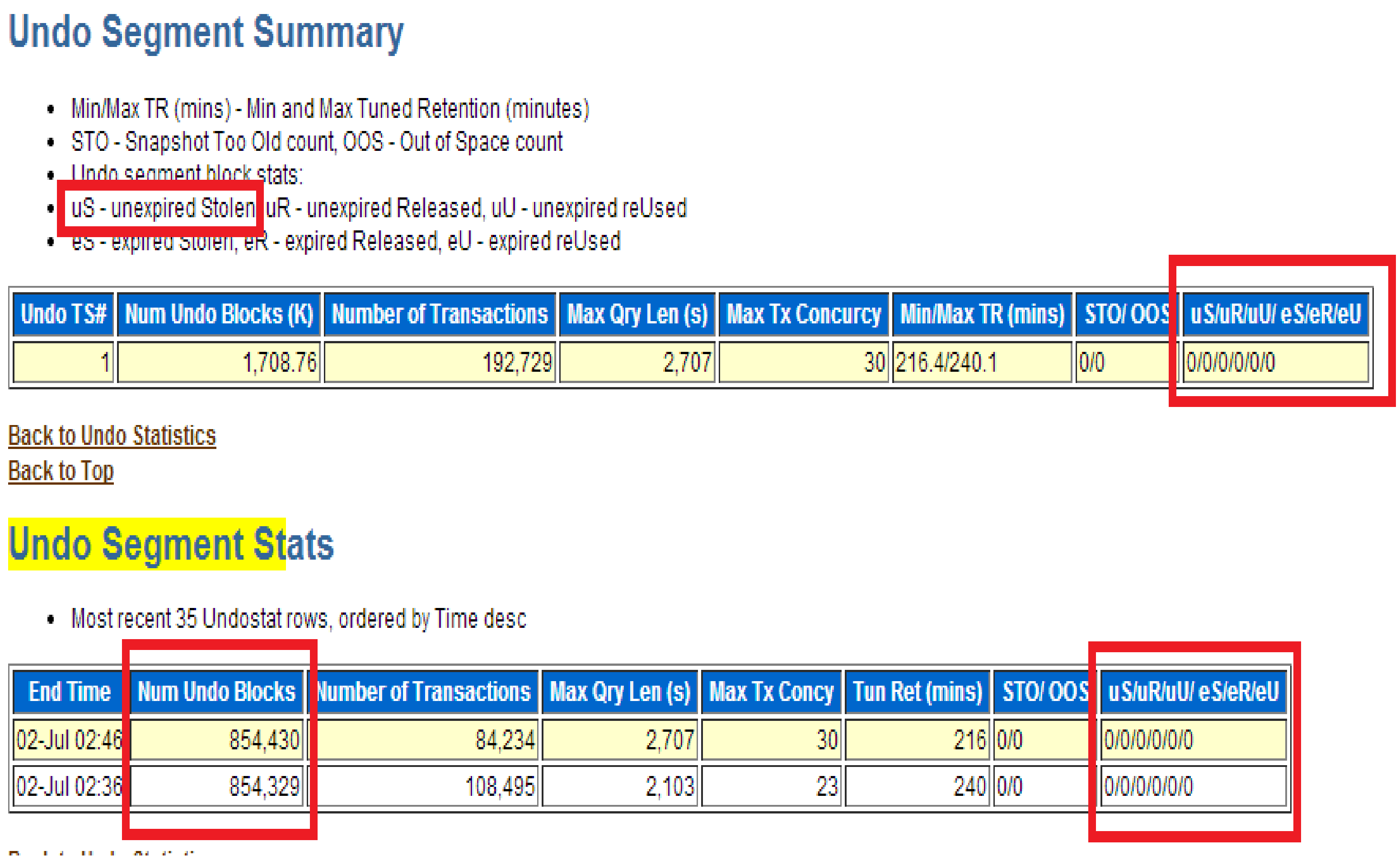


- Resources reported for PL/SQL code includes the resources used by all SQL statements called by the code.
- % Total DB Time is the Elapsed Time of the SQL statement divided into the Total Database Time multiplied by 100

Elapsed Time (s)	CPU Time (s	Executions	Elap per Exec (s)	% Total DB Time	SQL Id	SQL Module	SQL Text
20,006	6,518	3,083	6.49	49.97	1bwudbzpbdq5c	V3)	insert into tb_a (JIOYXH, JIO...
847	43	0		2.12	9as3jpq35tdhn	SQL*Plus	BEGIN proc_clean_tbl('BDPALH',...
847	302	0		2.12	c2mqbqr1d3vqh	SQL*Plus	BEGIN proc_clean_tbl('BJRNL', ...



SQL ID	Planhash	% Activity	Event	% Event	SQL Text
1bwudbzpbdq5c	750205943	50.33	CPU + Wait for CPU	32.21	insert into tba (JIOYXH, JIO...
			latch: undo global data	15.29	
			latch: cache buffers chains	2.08	
4n8ncj0uq01jq		12.20	CPU + Wait for CPU	7.37	** SQL Text Not Available **
		12.20	latch: undo global data	4.09	** SQL Text Not Available **
35qf7p1x57hqz	938792398	1.75	db file sequential read	1.44	delete from tb_a w...



Undo Segment Summary

- Min/Max TR (mins) - Min and Max Tuned Retention (minutes)
- STO - Snapshot Too Old count, OOS - Out of Space count
- Undo segment block stats:
- uS - unexpired Stolen, uR - unexpired Released, uU - unexpired reUsed
- eS - expired Stolen, eR - expired Released, eU - expired reUsed

Undo TS#	Num Undo Blocks (K)	Number of Transactions	Max Qry Len (s)	Max Tx Concurcy	Min/Max TR (mins)	STO/ OOS	uS/uR/uU/ eS/eR/eU
1	1,708.76	192,729	2,707	30	216.4/240.1	0/0	0/0/0/0/0/0

Back to Undo Statistics
Back to Top

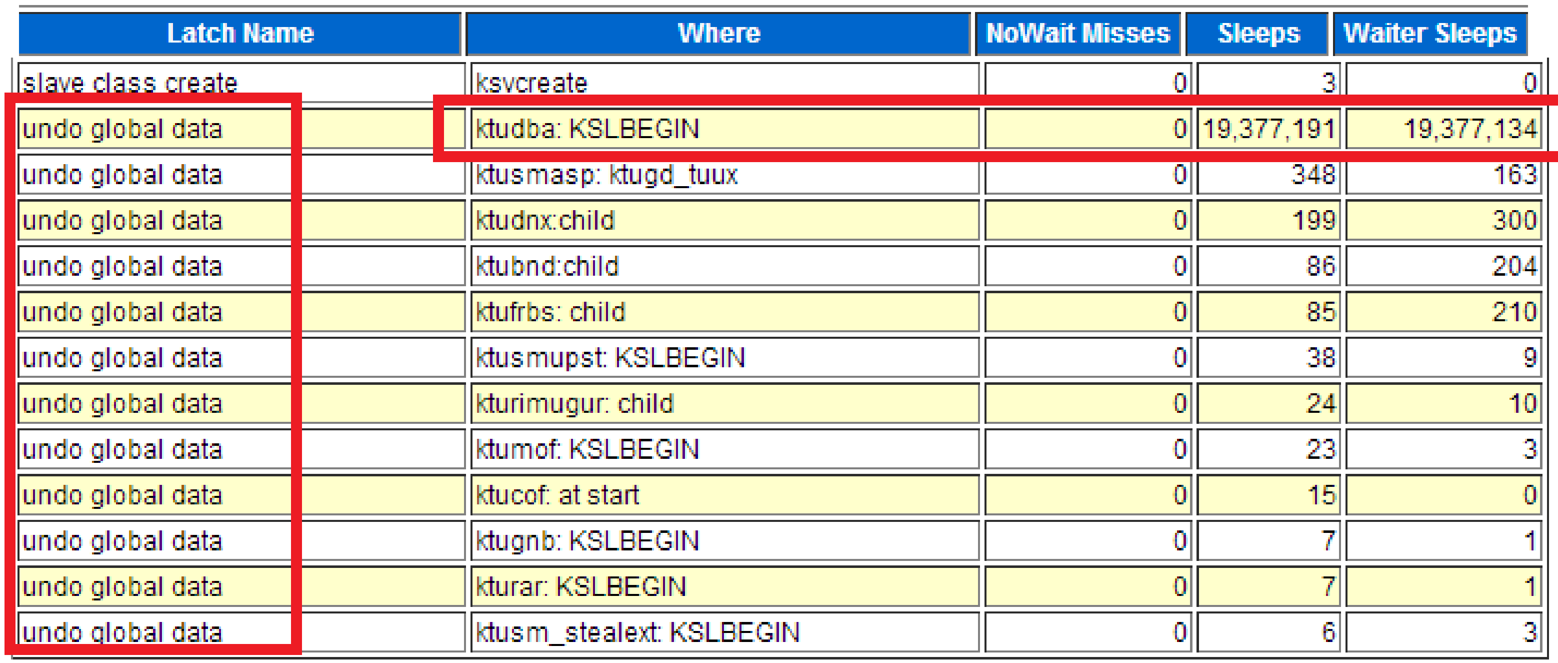
Undo Segment Stats

- Most recent 35 Undostat rows, ordered by Time desc

End Time	Num Undo Blocks	Number of Transactions	Max Qry Len (s)	Max Tx Concy	Tun Ret (mins)	STO/ OOS	uS/uR/uU/ eS/eR/eU
02-Jul 02:46	854,430	84,234	2,707	30	216	0/0	0/0/0/0/0/0
02-Jul 02:36	854,329	108,495	2,103	23	240	0/0	0/0/0/0/0/0

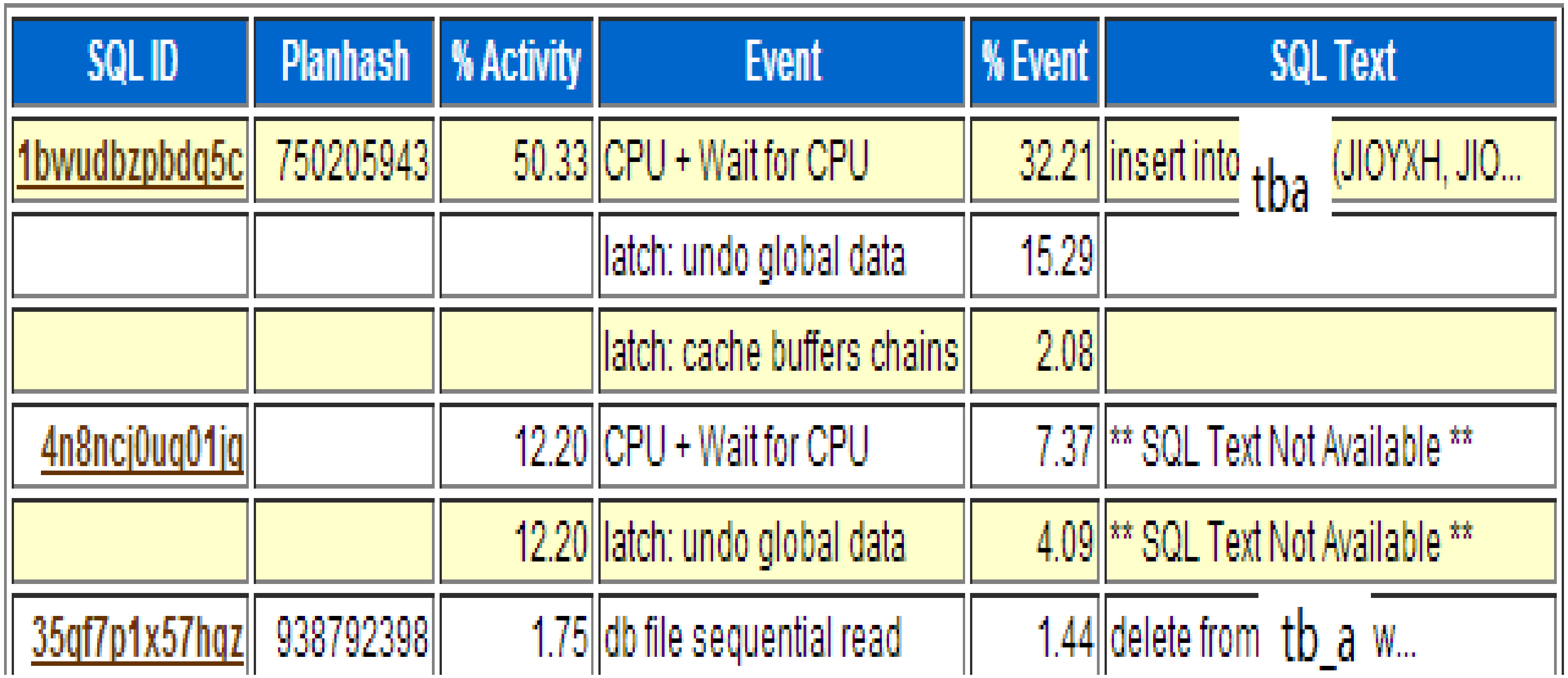
到底是什么原因呢？

- 先理解等待事件的本质原理



Latch Name	Where	NoWait Misses	Sleeps	Waiter Sleeps
slave class create	ksvcreate	0	3	0
undo global data	ktudba: KSLBEGIN	0	19,377,191	19,377,134
undo global data	ktusmasp: ktugd_tuux	0	348	163
undo global data	ktudnx:child	0	199	300
undo global data	ktubnd:child	0	86	204
undo global data	ktufrbs: child	0	85	210
undo global data	ktusmupst: KSLBEGIN	0	38	9
undo global data	kturimugur: child	0	24	10
undo global data	ktumof: KSLBEGIN	0	23	3
undo global data	ktucof: at start	0	15	0
undo global data	ktugnb: KSLBEGIN	0	7	1
undo global data	kturar: KSLBEGIN	0	7	1
undo global data	ktusm_stealext: KSLBEGIN	0	6	3

China Etek Service



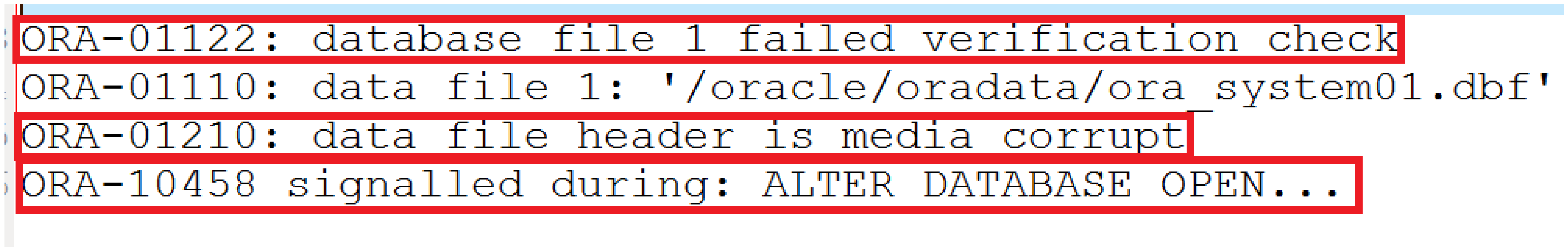
SQL ID	Planhash	% Activity	Event	% Event	SQL Text
1bwudbzpbdq5c	750205943	50.33	CPU + Wait for CPU	32.21	insert into tba (JIOYXH, JIO...
			latch: undo global data	15.29	
			latch: cache buffers chains	2.08	
4n8ncj0uq01jq		12.20	CPU + Wait for CPU	7.37	** SQL Text Not Available **
		12.20	latch: undo global data	4.09	** SQL Text Not Available **
35qf7p1x57hqz	938792398	1.75	db file sequential read	1.44	delete from tb_a w...

下一个案例

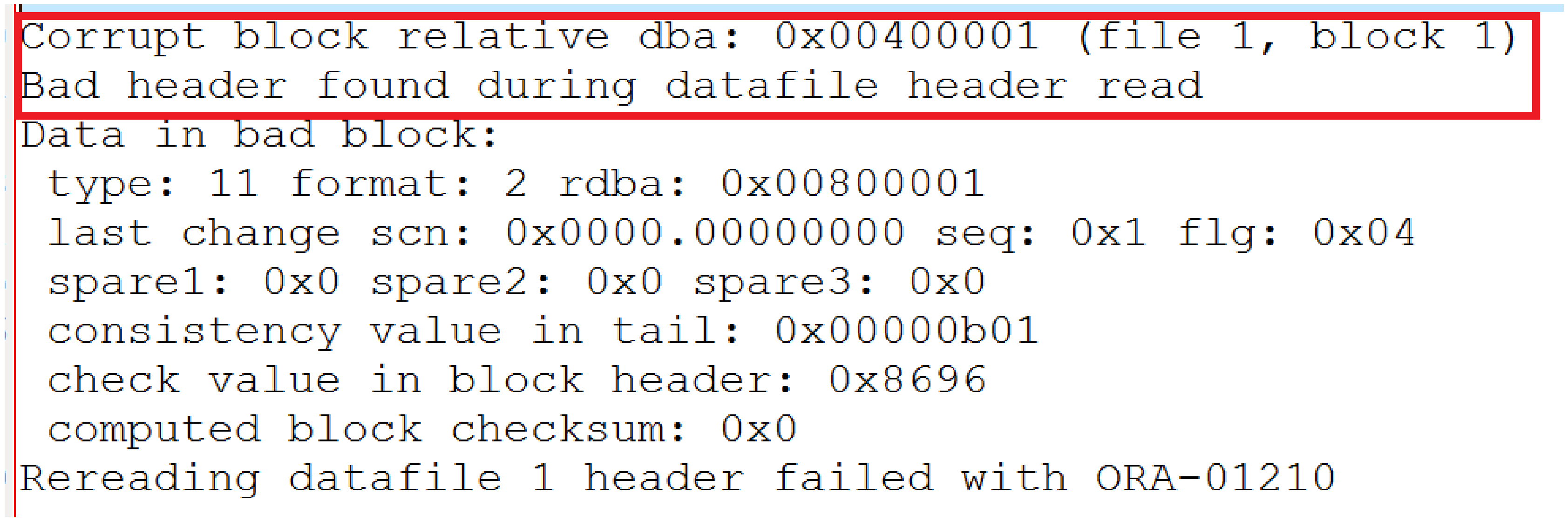
中亦科技
China Etek Service

Perfecting IT service and favoring clients 'success
锻造凝练IT服务 助推用户事业发展

问题-操作系统异常掉电后数据库无法启动



```
ORA-01122: database file 1 failed verification check
ORA-01110: data file 1: '/oracle/oradata/ora_system01.dbf'
ORA-01210: data file header is media corrupt
ORA-10458 signalled during: ALTER DATABASE OPEN...
```



```
Corrupt block relative dba: 0x00400001 (file 1, block 1)
Bad header found during datafile header read
Data in bad block:
 type: 11 format: 2 rdba: 0x00800001
 last change scn: 0x0000.00000000 seq: 0x1 flg: 0x04
 spare1: 0x0 spare2: 0x0 spare3: 0x0
 consistency value in tail: 0x00000b01
 check value in block header: 0x8696
 computed block checksum: 0x0
Rereading datafile 1 header failed with ORA-01210
```

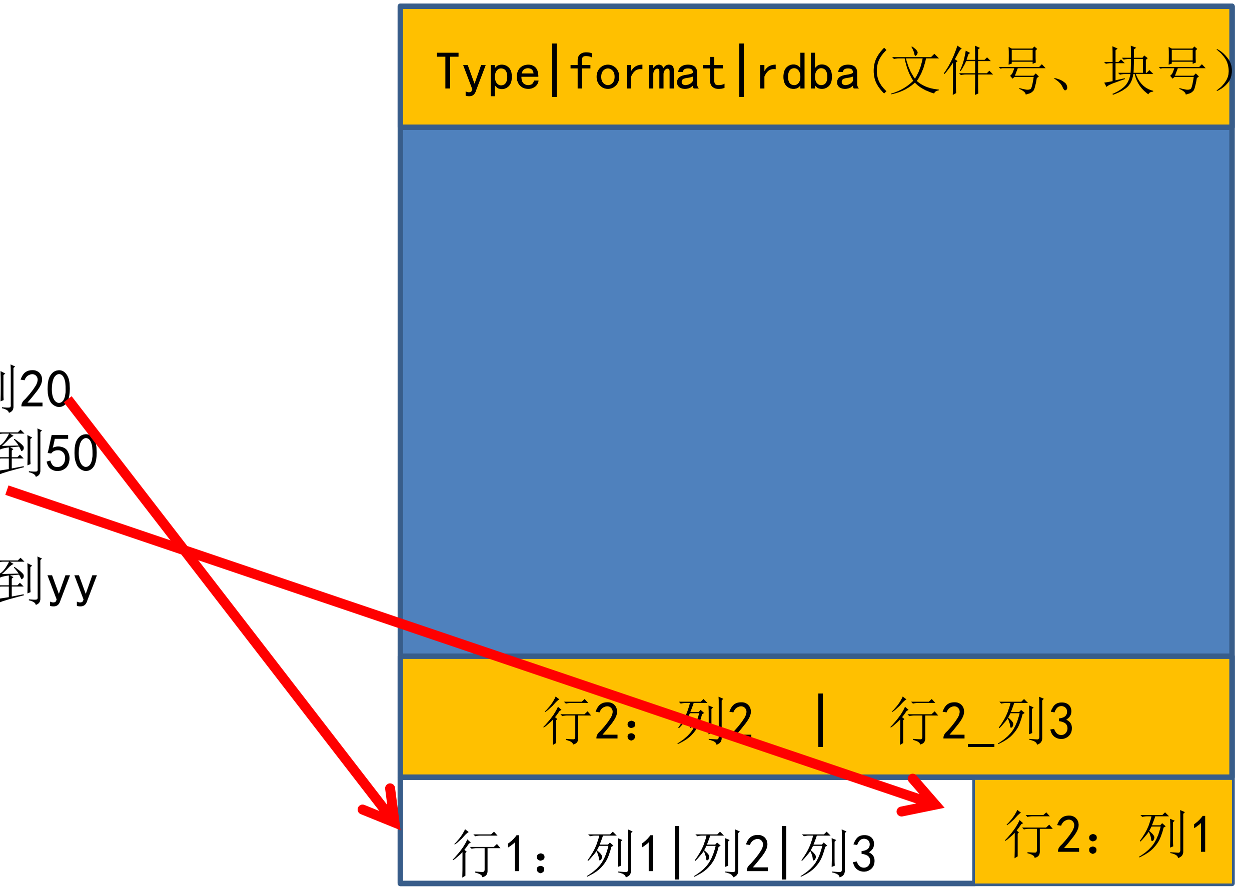
如果是你遇到这样的错误怎么办？

```
#########RDBA即数据块地址

十六进制                十进制
---------------      ---------------
0X00800001           8388609

#########RDBA转变成文件号、块号
select
dbms_utility.DATA_BLOCK_ADDRESS_FILE(8388609)  file_no,
dbms_utility.DATA_BLOCK_ADDRESS_BLOCK(8388609) block_no
from dual;

file_no    block_no
---------  ----------
   2           1
```

这个错误到底表示什么意思-数据块格式存储

中亦科技
China Etek Service

Row directory
第1行的offset 0到20
第2行的offset 20到50
…
第n行的offset xx到yy

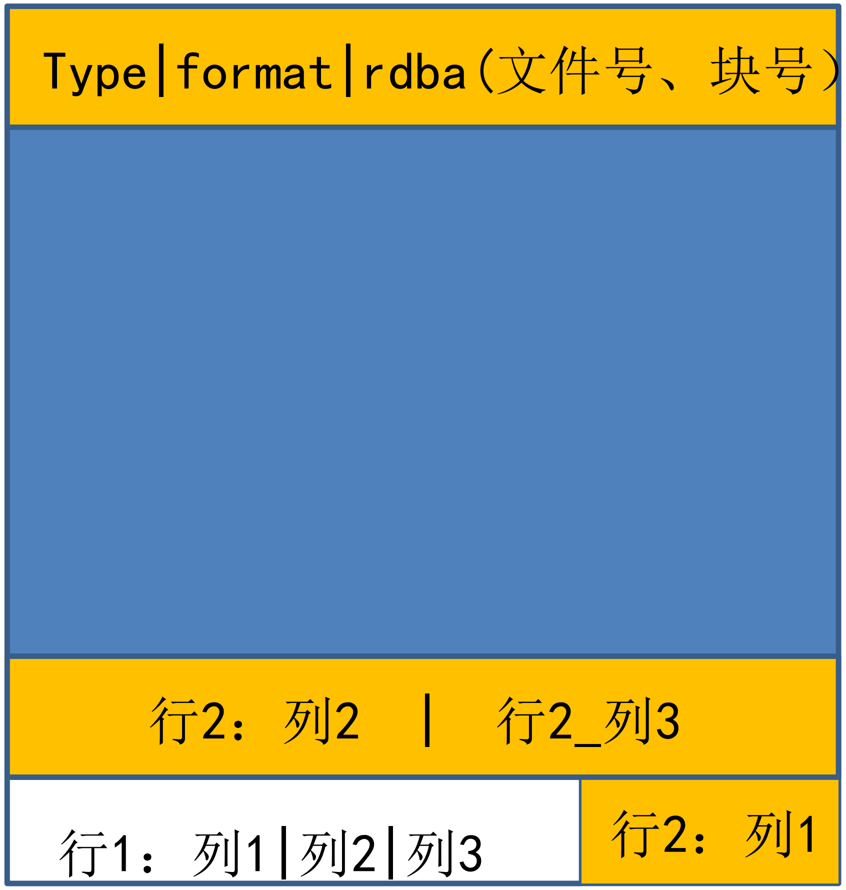


数据块格式存储

Oracle开始读取1号文件1号块，1号文件2号块

…

但是当读到1号文件1号块后，校验块的格式，发现第4到第7个字节的RDBA其实是2号文件1号块



遇到这样的问题要冷静！

中亦科技
China Etek Service

1、无法找到问题原因，则问题反复发生

2、无法找到问题原因，盲目进行处理，使得问题变得更糟糕

3、无法找到问题原因，使得问题处理更发杂，业务受影响的时间更长

环境介绍和客户反馈的信息

1、基础环境：

rhel x86_64bit

oracle 11.2.0.4.6

数据文件存放在裸设备上

2、备份情况：

无备份！

3、客户初步判断：

主机重启后，操作系统把数据文件的头给重写了！！

导致1号文件文件头损坏，继而数据库无法启动！

接下来如果是你，怎么处理？

客户的选择

由于无法查到原因

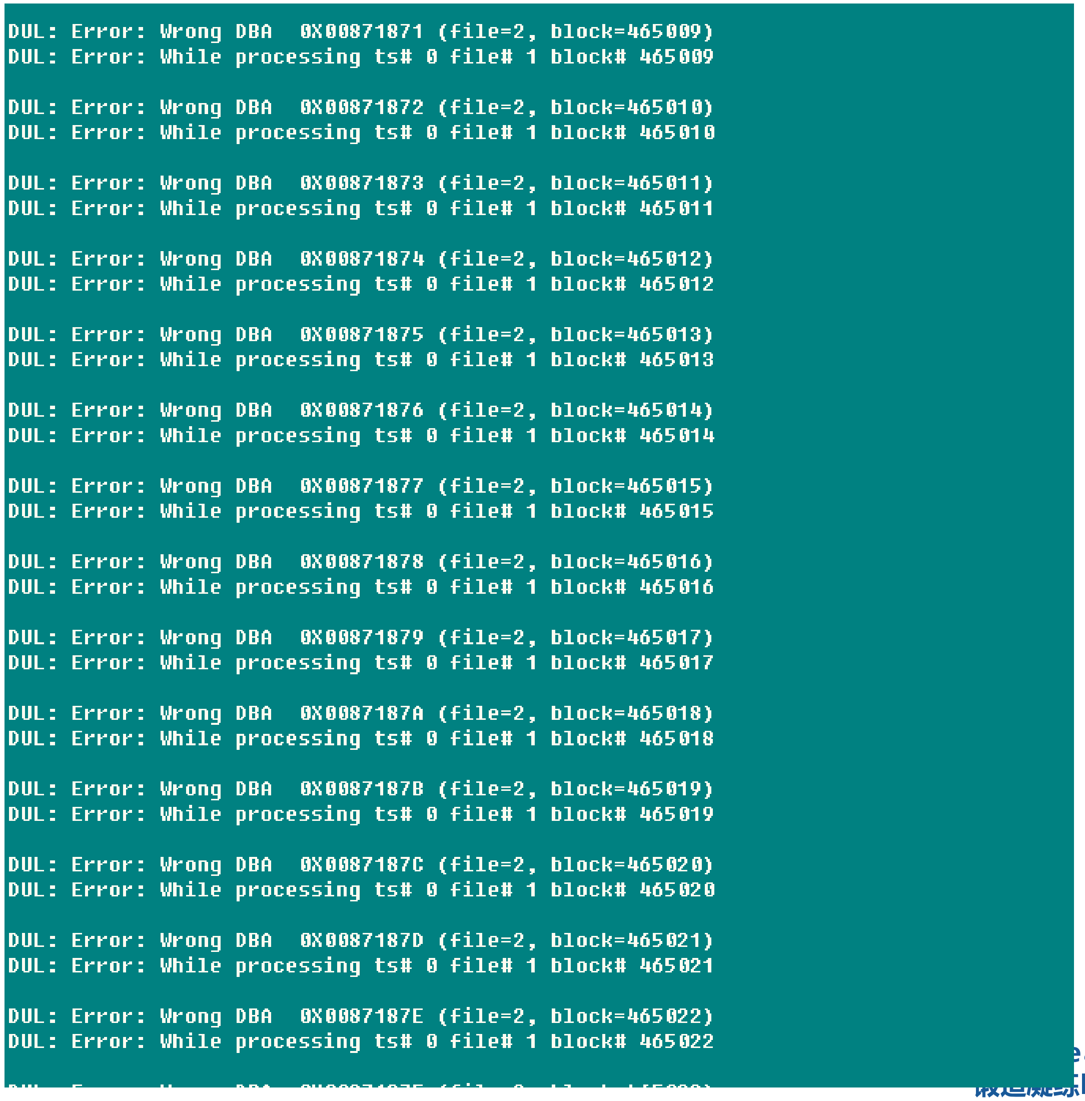
数据库又无法打开

所以客户开始准备使用DUL软件对数据文件进行抽取

（也有些客户选择设置隐含参数，强行拉库，后面你会发现完全不可行！）

DUL bootstrap直接抽数报错，求助我们

中亦科技
China Etek Service



```
DUL: Error: Wrong DBA  0X00871871 (file=2, block=465009)
DUL: Error: While processing ts# 0 file# 1 block# 465009

DUL: Error: Wrong DBA  0X00871872 (file=2, block=465010)
DUL: Error: While processing ts# 0 file# 1 block# 465010

DUL: Error: Wrong DBA  0X00871873 (file=2, block=465011)
DUL: Error: While processing ts# 0 file# 1 block# 465011

DUL: Error: Wrong DBA  0X00871874 (file=2, block=465012)
DUL: Error: While processing ts# 0 file# 1 block# 465012

DUL: Error: Wrong DBA  0X00871875 (file=2, block=465013)
DUL: Error: While processing ts# 0 file# 1 block# 465013

DUL: Error: Wrong DBA  0X00871876 (file=2, block=465014)
DUL: Error: While processing ts# 0 file# 1 block# 465014

DUL: Error: Wrong DBA  0X00871877 (file=2, block=465015)
DUL: Error: While processing ts# 0 file# 1 block# 465015

DUL: Error: Wrong DBA  0X00871878 (file=2, block=465016)
DUL: Error: While processing ts# 0 file# 1 block# 465016

DUL: Error: Wrong DBA  0X00871879 (file=2, block=465017)
DUL: Error: While processing ts# 0 file# 1 block# 465017

DUL: Error: Wrong DBA  0X0087187A (file=2, block=465018)
DUL: Error: While processing ts# 0 file# 1 block# 465018

DUL: Error: Wrong DBA  0X0087187B (file=2, block=465019)
DUL: Error: While processing ts# 0 file# 1 block# 465019

DUL: Error: Wrong DBA  0X0087187C (file=2, block=465020)
DUL: Error: While processing ts# 0 file# 1 block# 465020

DUL: Error: Wrong DBA  0X0087187D (file=2, block=465021)
DUL: Error: While processing ts# 0 file# 1 block# 465021

DUL: Error: Wrong DBA  0X0087187E (file=2, block=465022)
DUL: Error: While processing ts# 0 file# 1 block# 465022
```

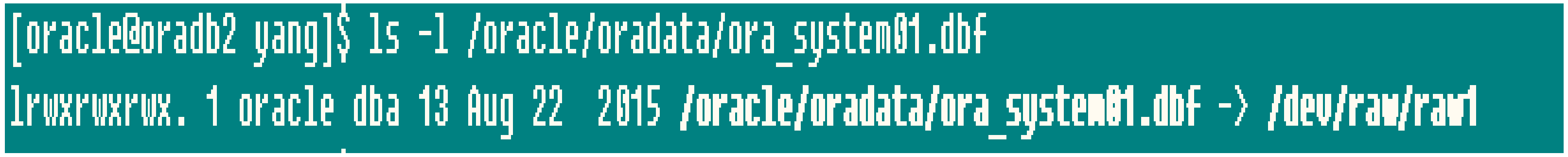
至此原因基本定位

中沶科技
China Etek Service

Perfecting IT service and favoring clients 'success
锻造凝练IT服务　助推用户事业发展

分析过程-梳理数据文件和裸设备的link关系

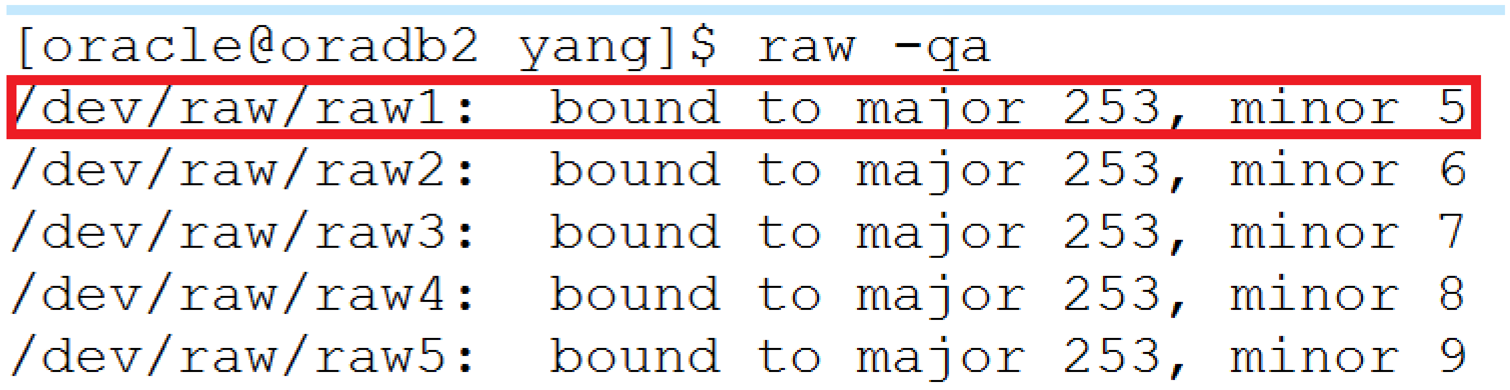
中亦科技
China Etek Service



```
[oracle@oradb2 yang]$ ls -l /oracle/oradata/ora_system01.dbf
lrwxrwxrwx. 1 oracle dba 13 Aug 22  2015 /oracle/oradata/ora_system01.dbf -> /dev/raw/raw1
```

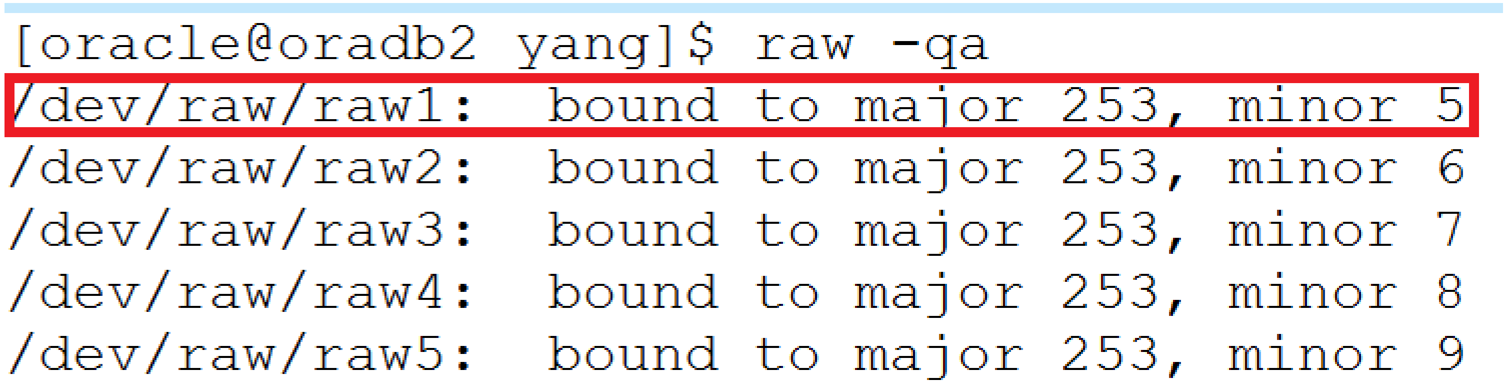
分析过程—梳理裸设备和块设备的映射关系

中亦科技
China Etek Service



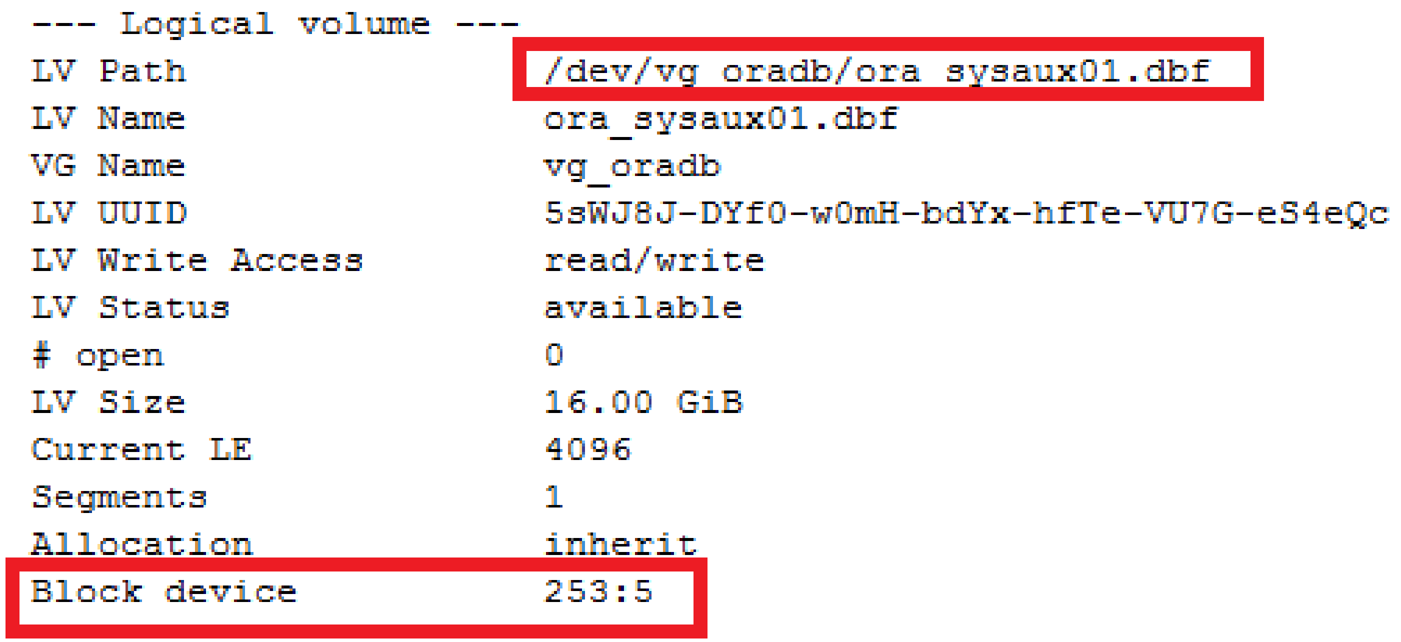
```
[oracle@oradb2 yang]$ raw -qa
/dev/raw/raw1:  bound to major 253, minor 5
/dev/raw/raw2:  bound to major 253, minor 6
/dev/raw/raw3:  bound to major 253, minor 7
/dev/raw/raw4:  bound to major 253, minor 8
/dev/raw/raw5:  bound to major 253, minor 9
```

分析过程



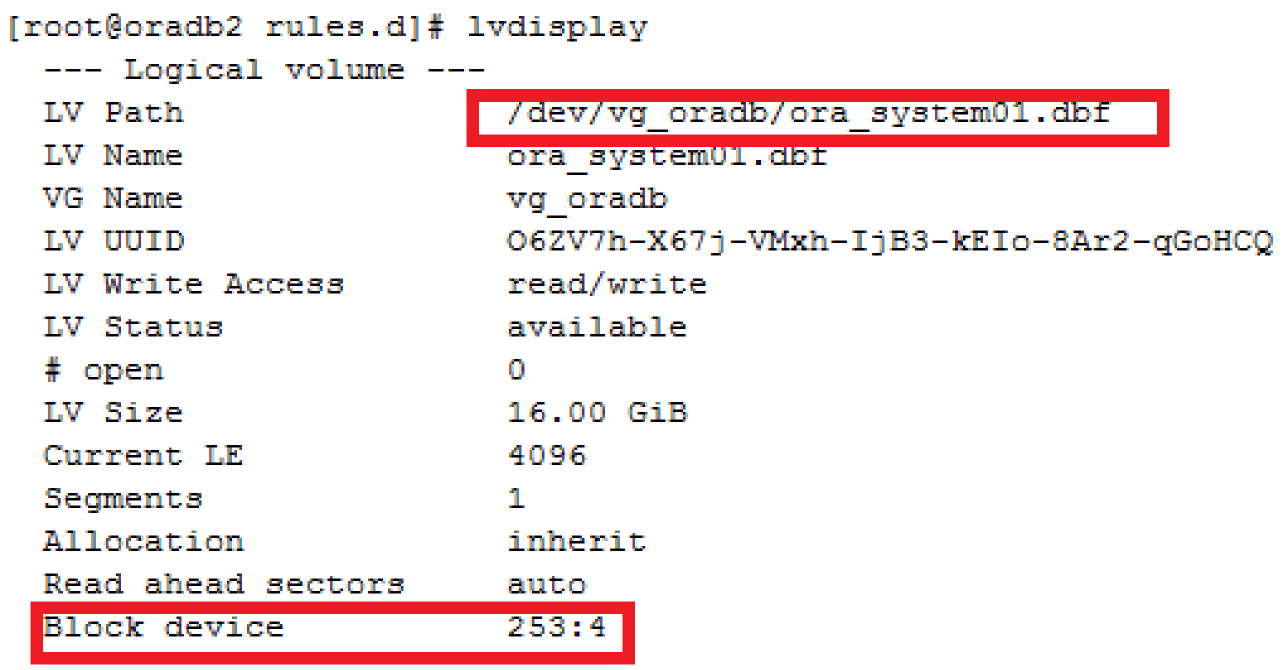
```
[oracle@oradb2 yang]$ raw -qa
/dev/raw/raw1:  bound to major 253, minor 5
/dev/raw/raw2:  bound to major 253, minor 6
/dev/raw/raw3:  bound to major 253, minor 7
/dev/raw/raw4:  bound to major 253, minor 8
/dev/raw/raw5:  bound to major 253, minor 9
```

开始发现异常



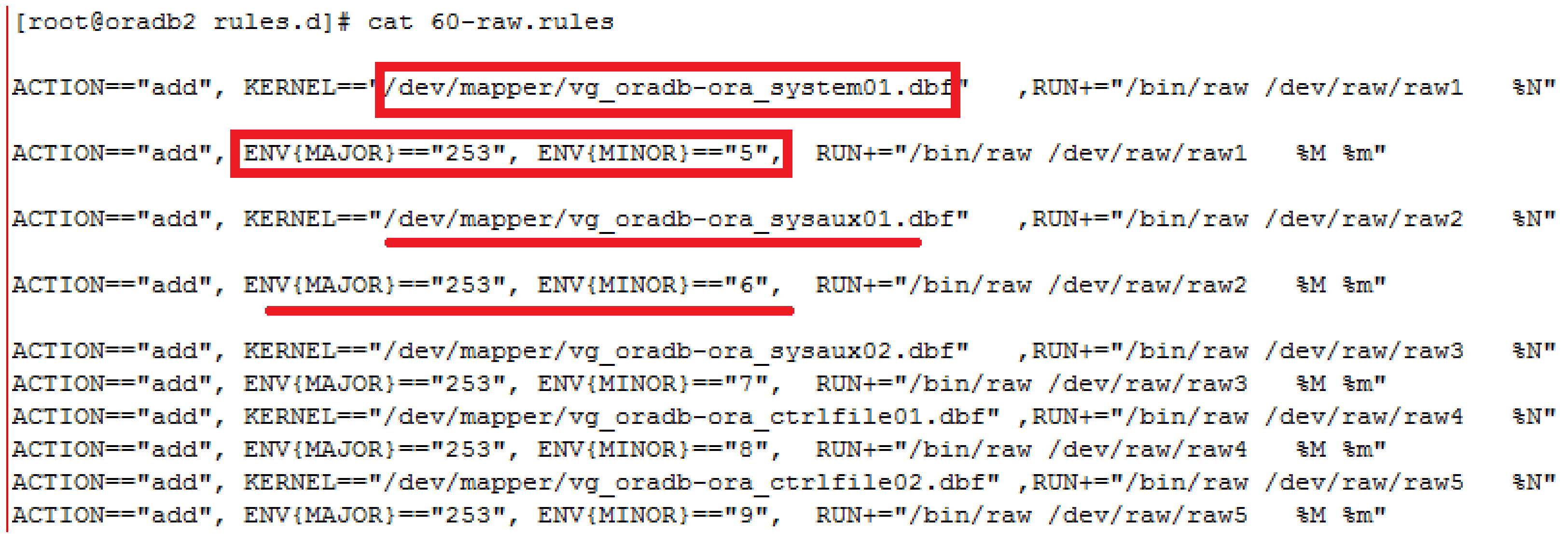
```
--- Logical volume ---
LV Path                /dev/vg_oradb/ora_sysaux01.dbf
LV Name                ora_sysaux01.dbf
VG Name                vg_oradb
LV UUID                5sWJ8J-DYf0-w0mH-bdYx-hfTe-VU7G-eS4eQc
LV Write Access        read/write
LV Status              available
# open                 0
LV Size                16.00 GiB
Current LE             4096
Segments               1
Allocation             inherit
Block device           253:5
```

异常数据（续）



```
[root@oradb2 rules.d]# lvdisplay
  --- Logical volume ---
  LV Path                /dev/vg_oradb/ora_system01.dbf
  LV Name                ora_system01.dbf
  VG Name                vg_oradb
  LV UUID                O6ZV7h-X67j-VMxh-IjB3-kEIo-8Ar2-qGoHCQ
  LV Write Access        read/write
  LV Status              available
  # open                 0
  LV Size                16.00 GiB
  Current LE             4096
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  Block device           253:4
```

为什么重启后问题出现



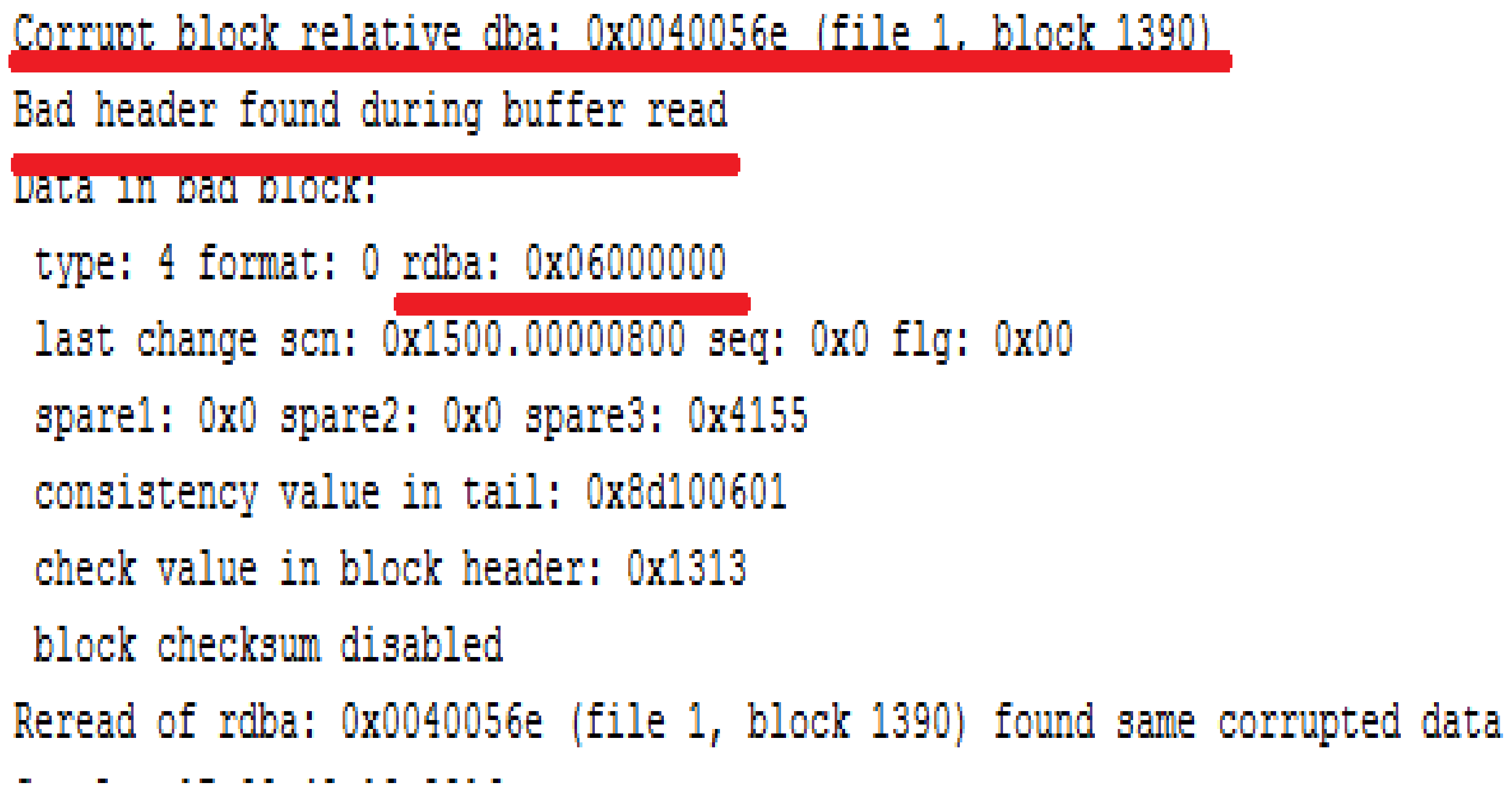
```
[root@oradb2 rules.d]# cat 60-raw.rules

ACTION=="add", KERNEL=="/dev/mapper/vg_oradb-ora_system01.dbf"   ,RUN+="/bin/raw /dev/raw/raw1   %N"

ACTION=="add", ENV{MAJOR}=="253", ENV{MINOR}=="5",  RUN+="/bin/raw /dev/raw/raw1   %M %m"

ACTION=="add", KERNEL=="/dev/mapper/vg_oradb-ora_sysaux01.dbf"   ,RUN+="/bin/raw /dev/raw/raw2   %N"

ACTION=="add", ENV{MAJOR}=="253", ENV{MINOR}=="6",  RUN+="/bin/raw /dev/raw/raw2   %M %m"

ACTION=="add", KERNEL=="/dev/mapper/vg_oradb-ora_sysaux02.dbf"   ,RUN+="/bin/raw /dev/raw/raw3   %N"
ACTION=="add", ENV{MAJOR}=="253", ENV{MINOR}=="7",  RUN+="/bin/raw /dev/raw/raw3   %M %m"
ACTION=="add", KERNEL=="/dev/mapper/vg_oradb-ora_ctrlfile01.dbf" ,RUN+="/bin/raw /dev/raw/raw4   %N"
ACTION=="add", ENV{MAJOR}=="253", ENV{MINOR}=="8",  RUN+="/bin/raw /dev/raw/raw4   %M %m"
ACTION=="add", KERNEL=="/dev/mapper/vg_oradb-ora_ctrlfile02.dbf" ,RUN+="/bin/raw /dev/raw/raw5   %N"
ACTION=="add", ENV{MAJOR}=="253", ENV{MINOR}=="9",  RUN+="/bin/raw /dev/raw/raw5   %M %m"
```

下一个案例

中亦科技
China Etek Service

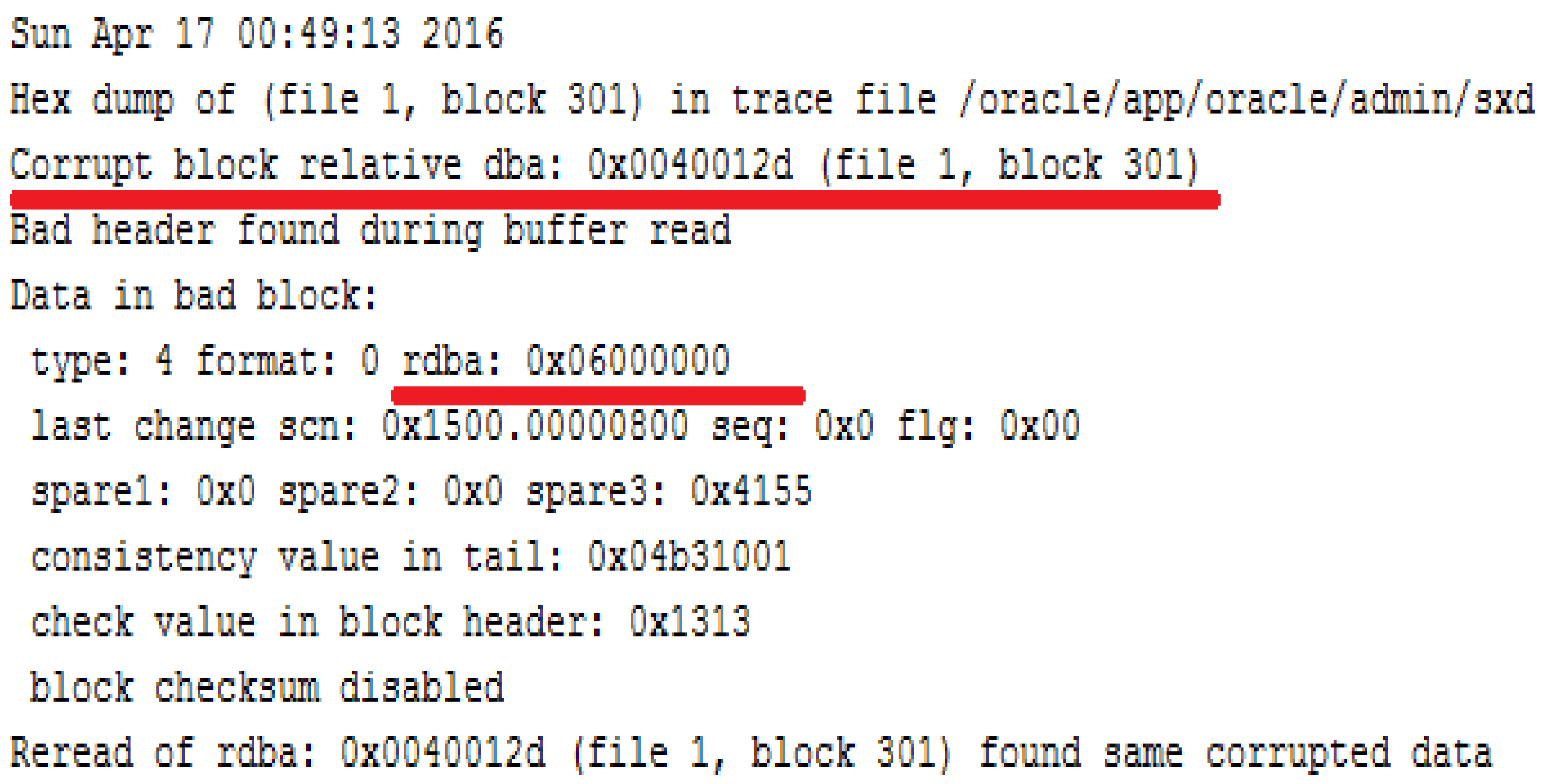
4月刚出现的故障（system数据文件出现坏块）

中亦科技
China Etek Service



```
Corrupt block relative dba: 0x0040056e (file 1, block 1390)
Bad header found during buffer read
Data in bad block:
 type: 4 format: 0 rdba: 0x06000000
 last change scn: 0x1500.00000800 seq: 0x0 flg: 0x00
 spare1: 0x0 spare2: 0x0 spare3: 0x4155
 consistency value in tail: 0x8d100601
 check value in block header: 0x1313
 block checksum disabled
Reread of rdba: 0x0040056e (file 1, block 1390) found same corrupted data
```

续



```
Sun Apr 17 00:49:13 2016
Hex dump of (file 1, block 301) in trace file /oracle/app/oracle/admin/sxd
Corrupt block relative dba: 0x0040012d (file 1, block 301)
Bad header found during buffer read
Data in bad block:
 type: 4 format: 0 rdba: 0x06000000
 last change scn: 0x1500.00000800 seq: 0x0 flg: 0x00
 spare1: 0x0 spare2: 0x0 spare3: 0x4155
 consistency value in tail: 0x04b31001
 check value in block header: 0x1313
 block checksum disabled
Reread of rdba: 0x0040012d (file 1, block 301) found same corrupted data
```

如果是你，你会怎么处理

中亦科技
China Etek Service

DBV校验整个数据库，只有system表空间1号文件中的2个BLOCK是损坏的！

重启数据库？

千万别！！会让事情变得更复杂。

一旦重启后起不来，那么很多手段都用不了！

悲剧发生了

中亦科技
China Etek Service

1、Rman blockrecover 来恢复？

2、数据量大小为20T，没有rman备份和逻辑导出备份

3、因为数据量大，所以没有备份？

4、数据重要么？重要！那为什么…

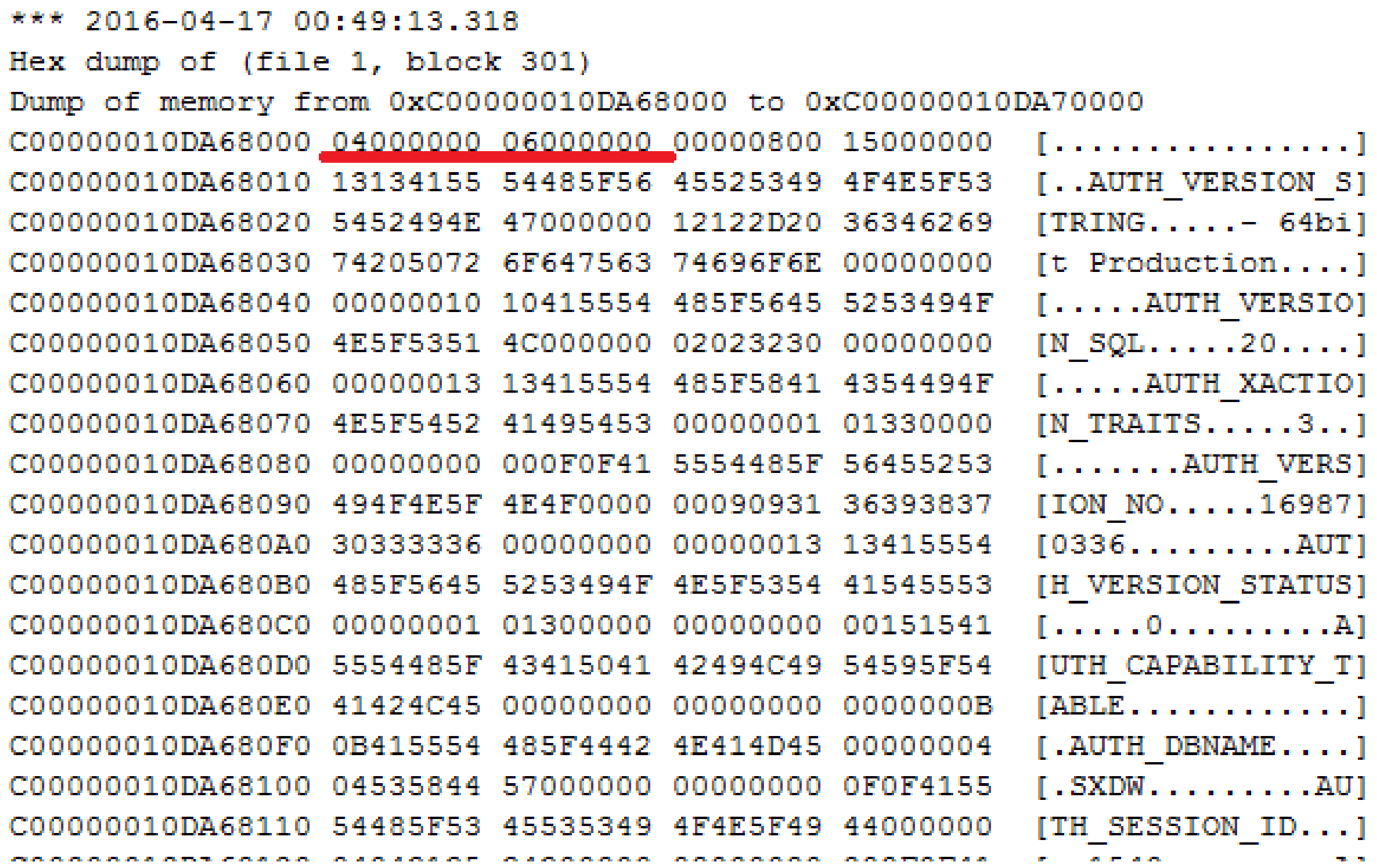
5、还有救么？损坏的是1号文件，即system表空间的数据文件，而不是用户数据，此处损坏的是非常重要的obj$基表，因此可以采取数据抽取软件进行恢复，时间可能需要1周以上！

6、还有更好的方法么？ Obj$ 基表数据块损坏！

思考一下，如何判断坏块成因？

中亦科技
China Etek Service

如何分析原因（坏块trace)



```
*** 2016-04-17 00:49:13.318
Hex dump of (file 1, block 301)
Dump of memory from 0xC00000010DA68000 to 0xC00000010DA70000
C00000010DA68000 04000000 06000000 00000800 15000000  [................]
C00000010DA68010 13134155 54485F56 45525349 4F4E5F53  [..AUTH_VERSION_S]
C00000010DA68020 5452494E 47000000 12122D20 36346269  [TRING.....- 64bi]
C00000010DA68030 74205072 6F647563 74696F6E 00000000  [t Production....]
C00000010DA68040 00000010 10415554 485F5645 5253494F  [.....AUTH_VERSIO]
C00000010DA68050 4E5F5351 4C000000 02023230 00000000  [N_SQL.....20....]
C00000010DA68060 00000013 13415554 485F5841 4354494F  [.....AUTH_XACTIO]
C00000010DA68070 4E5F5452 41495453 00000001 01330000  [N_TRAITS.....3..]
C00000010DA68080 00000000 000F0F41 5554485F 56455253  [.......AUTH_VERS]
C00000010DA68090 494F4E5F 4E4F0000 00090931 36393837  [ION_NO.....16987]
C00000010DA680A0 30333336 00000000 00000013 13415554  [0336.........AUT]
C00000010DA680B0 485F5645 5253494F 4E5F5354 41545553  [H_VERSION_STATUS]
C00000010DA680C0 00000001 01300000 00000000 00151541  [.....0.........A]
C00000010DA680D0 5554485F 43415041 42494C49 54595F54  [UTH_CAPABILITY_T]
C00000010DA680E0 41424C45 00000000 00000000 0000000B  [ABLE............]
C00000010DA680F0 0B415554 485F4442 4E414D45 00000004  [.AUTH_DBNAME....]
C00000010DA68100 04535844 57000000 00000000 0F0F4155  [.SXDW.........AU]
C00000010DA68110 54485F53 45535349 4F4E5F49 44000000  [TH_SESSION_ID...]
```

Perfecting IT service and favoring clients 'success
锻造凝练IT服务 助推用户事业发展

至此，真相浮出（崩溃...)

中亦科技
China Etek Service

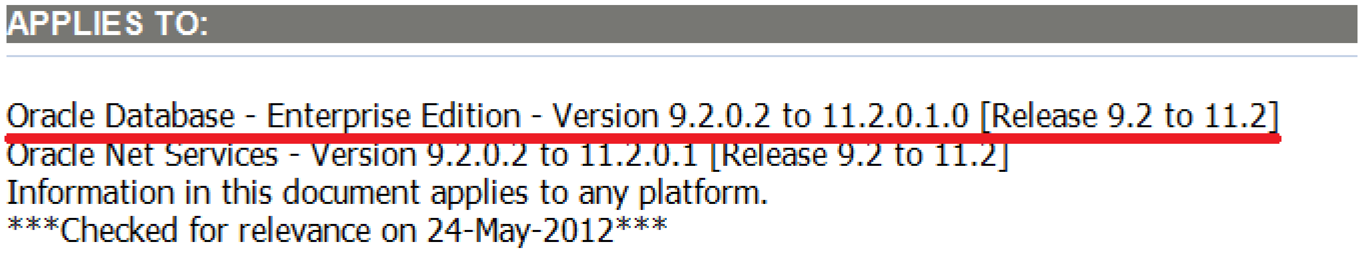
大家思考下，看看是否找到问题原因

坏块根因分析

ALERT: Bug 8943287 ORA-1578 corrupt block with SQL*Net AUTH strings (Doc ID 976852.1)

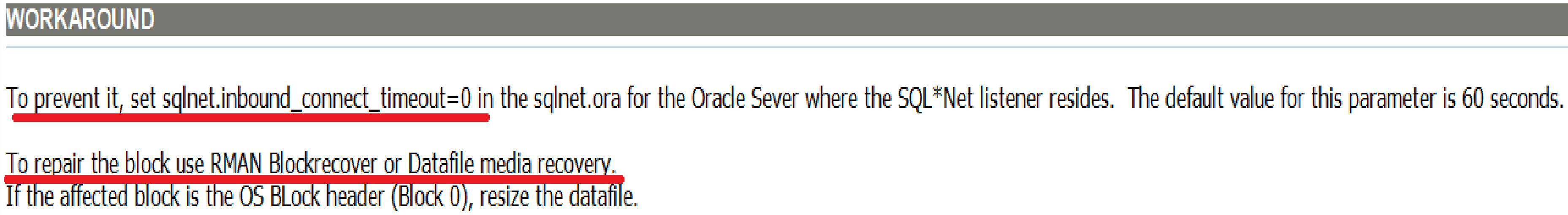
SQL*Net may overwrite database files with sqlnet packets containing AUTH strings when sqlnet.inbound_connect_timeout > 0. Subsequent SQL statements produce error ORA-1578.

如何预防



APPLIES TO:

Oracle Database - Enterprise Edition - Version 9.2.0.2 to 11.2.0.1.0 [Release 9.2 to 11.2]
Oracle Net Services - Version 9.2.0.2 to 11.2.0.1 [Release 9.2 to 11.2]
Information in this document applies to any platform.
Checked for relevance on 24-May-2012



WORKAROUND

To prevent it, set sqlnet.inbound_connect_timeout=0 in the sqlnet.ora for the Oracle Sever where the SQL*Net listener resides. The default value for this parameter is 60 seconds.

To repair the block use RMAN Blockrecover or Datafile media recovery.
If the affected block is the OS BLock header (Block 0), resize the datafile.

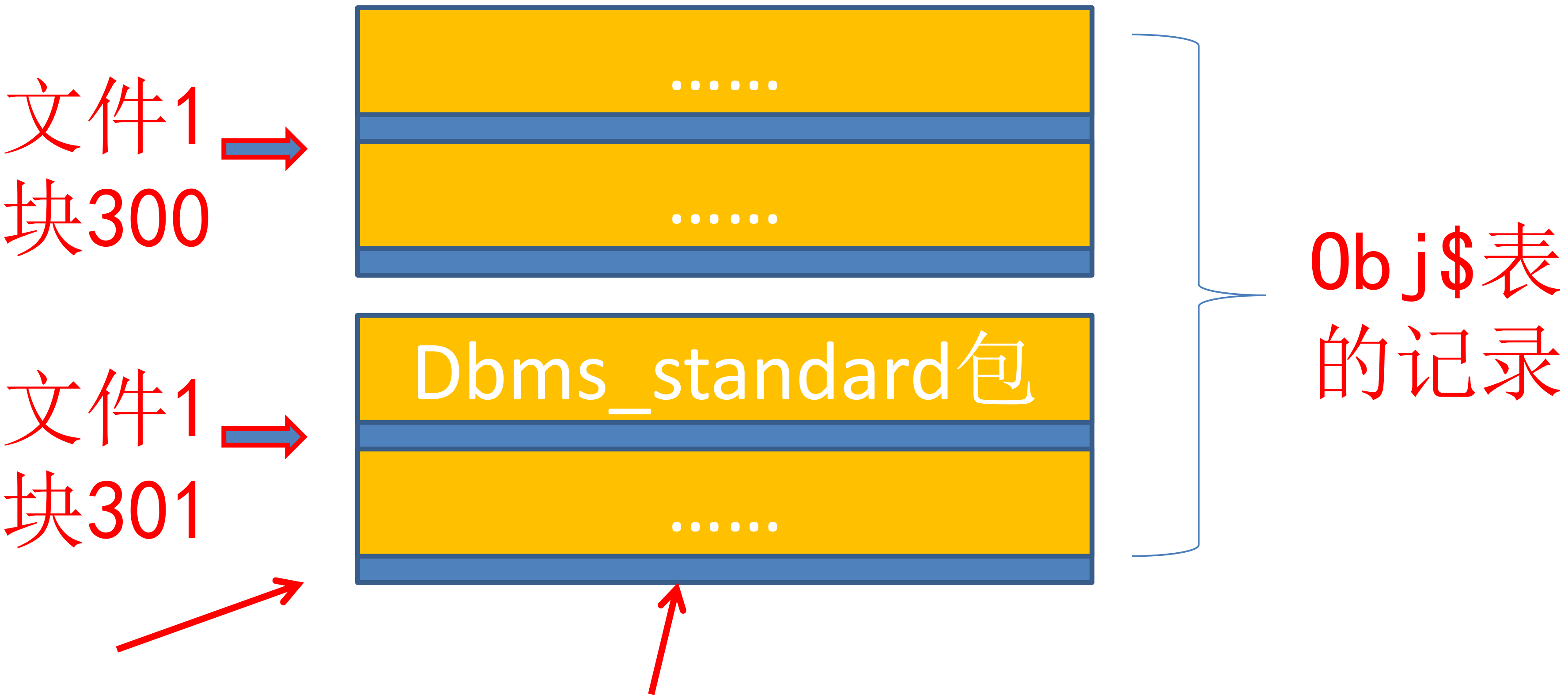
如何解决

中亦科技
China Etek Service

系统表空间坏，但是里面没有用户数据，并且数据库还是open状态

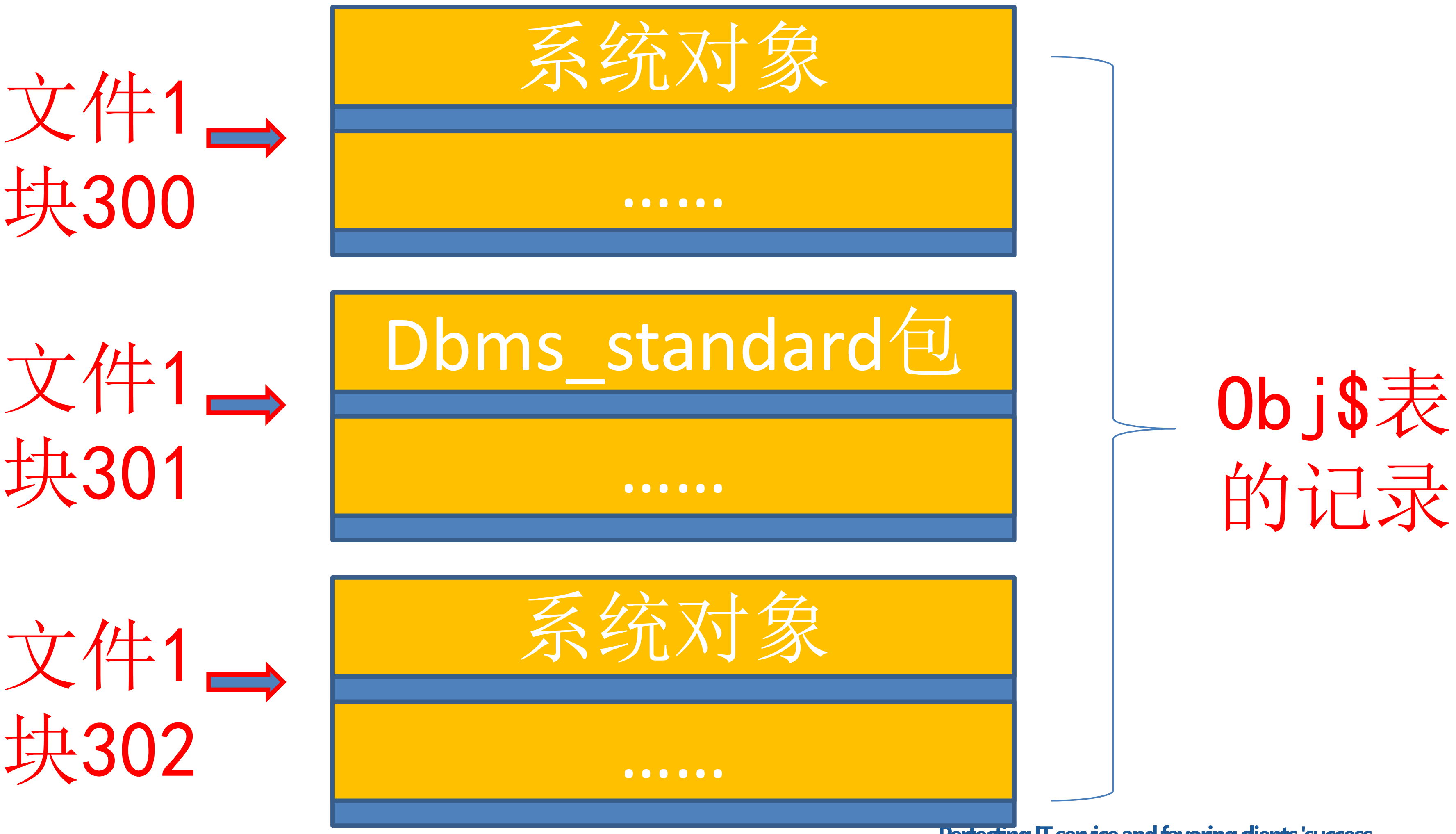
是不是跳过这些坏块（10231），然后导出数据，在新的环境重建库？

导出重建方式不可行



EXP/expdp报错，因为要调用dbms_start包，但这个对象的定义存在一个坏块里！

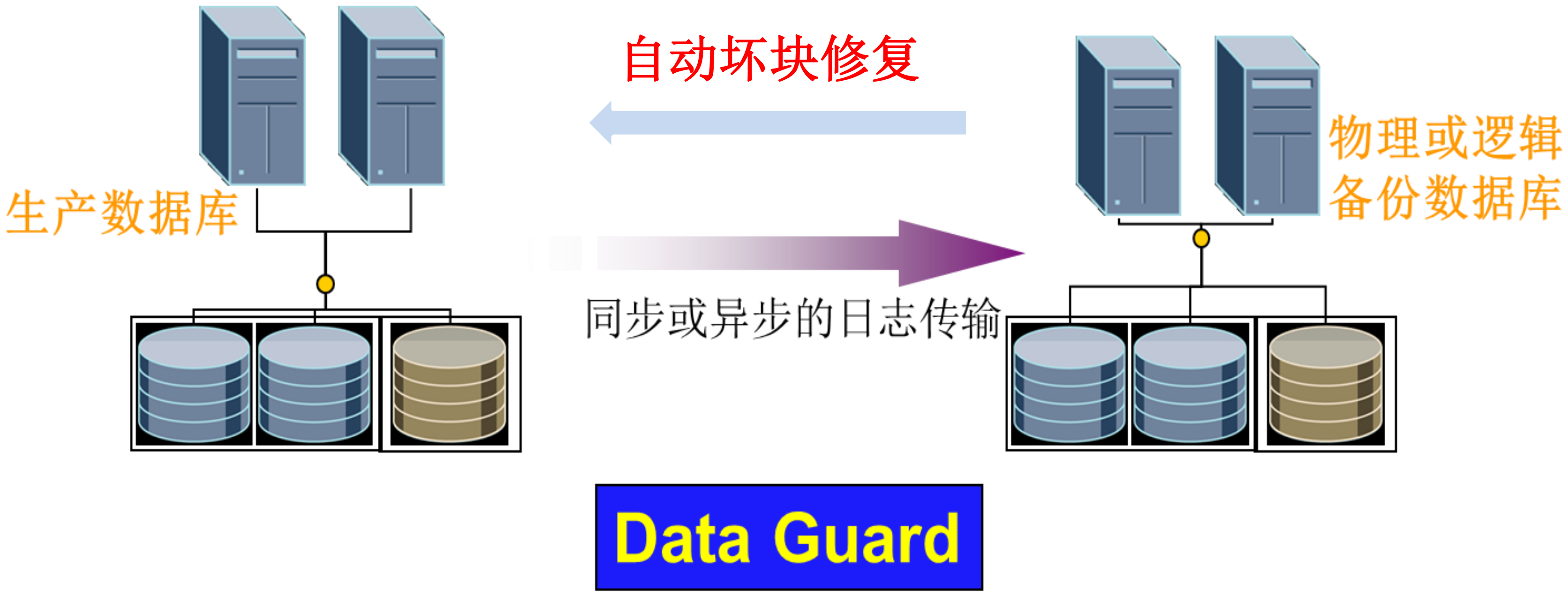
最终解决方案



如何通过数据库架构预防

1、通过磁带中的rman备份进行坏块修复的时间成本太高，无法满足业务连续性要求

2、通过active DataGuard可以自动修复坏块！



下一个案例

中亦科技
China Etek Service

Perfecting IT service and favoring clients 'success
锻造凝练IT服务　助推用户事业发展

案例-学无止境

一点体会：
命令是辅助的手段
思路到了，问题就定位和解决了
找到根因（root cause)后解决/绕开问题的方式就多了

案例介绍：
RAC中发生OS自动重启是必须学会处理的问题
Oswatcher会让问题变得简单
系统Hang主，没来得及写日志和信息

下面介绍如何使用kdb分析OS sysdump
以确定是否是由ORACLE CRS重启的一些技巧

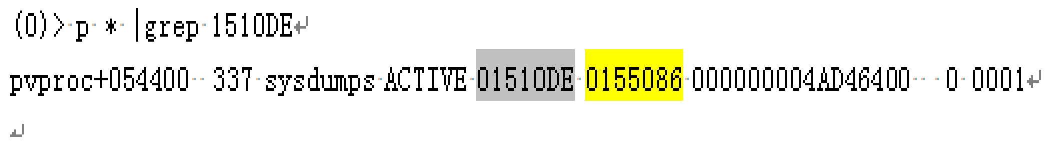
通过status查看CPU的状态，如果可以找到某科CPU正在做sysdumpstart，则有可能是10g RAC机制将OS重启。

```
 (0)> status
CPU     TID  TSLOT    PID  PSLOT  PROC_NAME
  0     2005      2   2004      2  wait
  1    10021     16   D01A     13  wait
  2    AE09F    174 1510DE    337  sysdumpstart <---
  3    13027     19   F01E     15  wait
  4    14029     20  10020     16  wait
  5    1502B     21  11022     17  wait
  6    91085    145 134032    308  oracle
  7    1702F     23  13026     19  wait
  8-63   Disabled
```

引起系统宕机的进程是sysdumpstart。这个进程一旦被调用，将会生成系统dump并重新启动系统

从黄色底纹部分获取到sysdumpstart进程号。

使用P * 命令来查看进程的信息

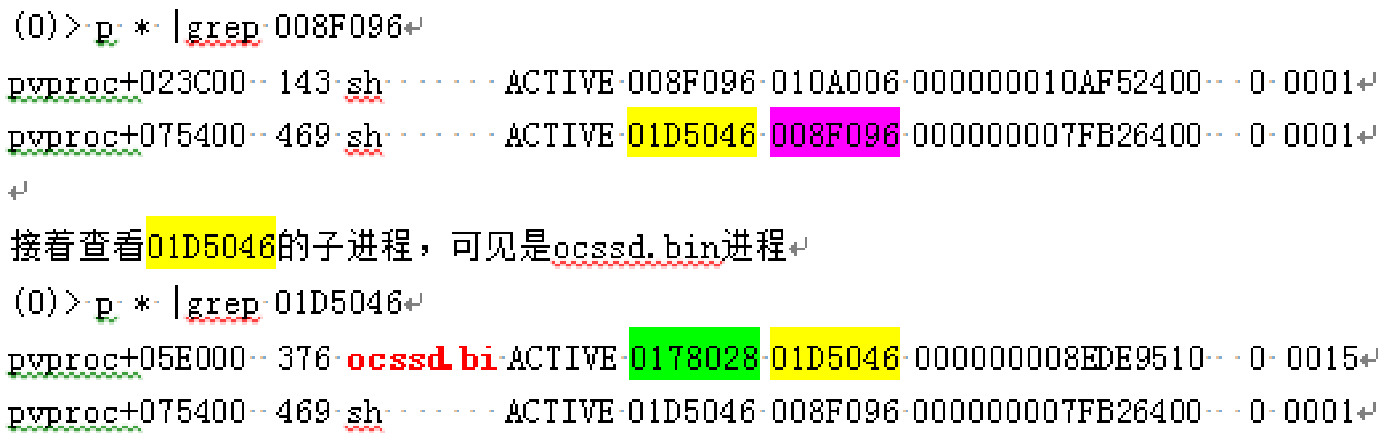


```
(0)> p * |grep 1510DE
pvproc+054400   337 sysdumps ACTIVE 01510DE 0155086 000000004AD46400    0 0001
```

灰色表示该进程的进程号，黄色表示父进程的进程号，用十六进程表示.上述的各列为

SLOT NAME   STATE PID   PPID   ……

同样的方法，继续把进程数缕出来



```
(0)> p * |grep 008F096
pvproc+023C00  143 sh       ACTIVE 008F096 010A006 000000010AF52400   0 0001
pvproc+075400  469 sh       ACTIVE 01D5046 008F096 000000007FB26400   0 0001
```

接着查看01D5046的子进程，可见是ocssd.bin进程

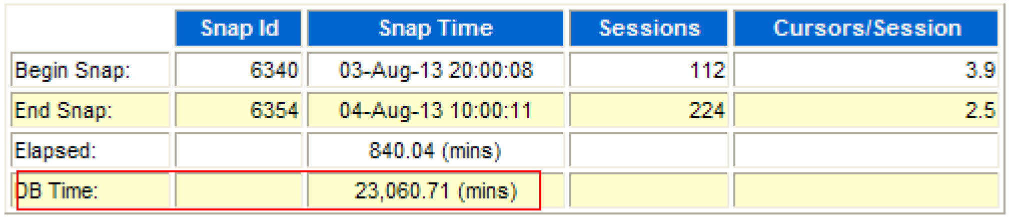
```
(0)> p * |grep 01D5046
pvproc+05E000  376 ocssd.bi ACTIVE 0178028 01D5046 000000008EDE9510   0 0015
pvproc+075400  469 sh       ACTIVE 01D5046 008F096 000000007FB26400   0 0001
```

从上面的信息可以看到：
sysdumpstart进程和ocssd.bin的进程具有相同的父进程(进程号010A006)，该父进程是一个shell程序，而ocssd.bin是oracle clusterware中的一个进程，因此该父进程必然是Oracle的一个脚本程序。Oracle的某个监控进程发现异常后调用了sysdumpstart让系统重新启动，以便维护集群的一致性。

下一个案例

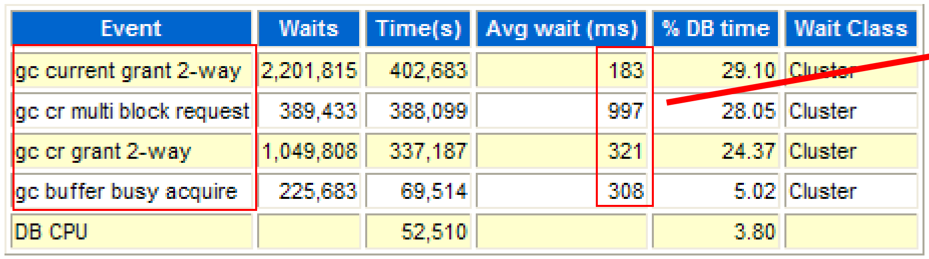
优化案例之数据库实例调优

- 现象
  - 执行计划完全一样情况下，批量时快时慢
  - 正常2小时，慢则长达10小时
  - 根据时间分解原则，时间主要消耗在RAC globalcache环节



	Snap Id	Snap Time	Sessions	Cursors/Session
Begin Snap:	6340	03-Aug-13 20:00:08	112	3.9
End Snap:	6354	04-Aug-13 10:00:11	224	2.5
Elapsed:		840.04 (mins)		
DB Time:		23,060.71 (mins)		

Top 5 Timed Foreground Events

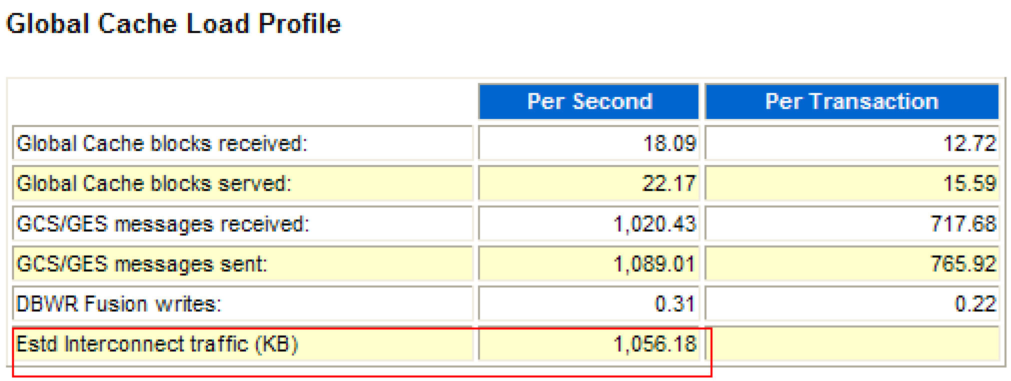


Event	Waits	Time(s)	Avg wait (ms)	% DB time	Wait Class
gc current grant 2-way	2,201,815	402,683	183	29.10	Cluster
gc cr multi block request	389,433	388,099	997	28.05	Cluster
gc cr grant 2-way	1,049,808	337,187	321	24.37	Cluster
gc buffer busy acquire	225,683	69,514	308	5.02	Cluster
DB CPU		52,510		3.80	

Gc单次等待达到几百毫秒！

优化案例之数据库实例调优

- 私网流量仅1M/秒，时间都消耗到哪了？

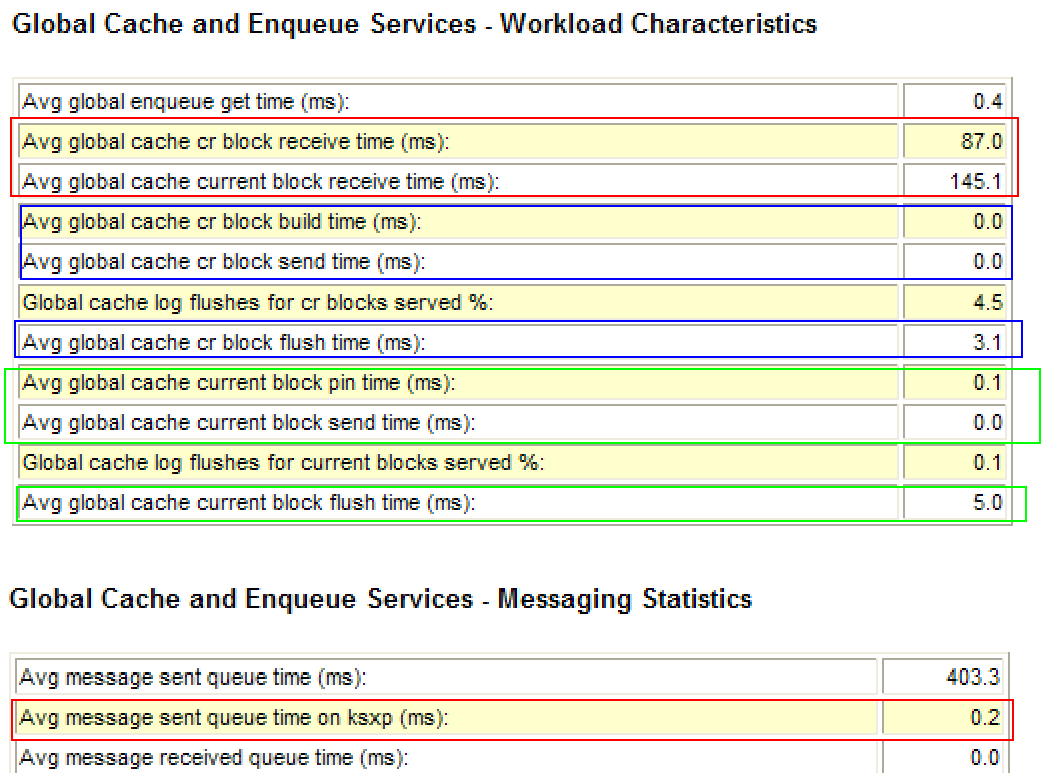


Global Cache Load Profile

	Per Second	Per Transaction
Global Cache blocks received:	18.09	12.72
Global Cache blocks served:	22.17	15.59
GCS/GES messages received:	1,020.43	717.68
GCS/GES messages sent:	1,089.01	765.92
DBWR Fusion writes:	0.31	0.22
Estd Interconnect traffic (KB)	1,056.18	

优化案例之数据库实例调优

- 对Global cache各个环节的时间进一步分解



Global Cache and Enqueue Services - Workload Characteristics

Avg global enqueue get time (ms):	0.4
Avg global cache cr block receive time (ms):	87.0
Avg global cache current block receive time (ms):	145.1
Avg global cache cr block build time (ms):	0.0
Avg global cache cr block send time (ms):	0.0
Global cache log flushes for cr blocks served %:	4.5
Avg global cache cr block flush time (ms):	3.1
Avg global cache current block pin time (ms):	0.1
Avg global cache current block send time (ms):	0.0
Global cache log flushes for current blocks served %:	0.1
Avg global cache current block flush time (ms):	5.0

Global Cache and Enqueue Services - Messaging Statistics

Avg message sent queue time (ms):	403.3
Avg message sent queue time on ksxp (ms):	0.2
Avg message received queue time (ms):	0.0

按照公式计算

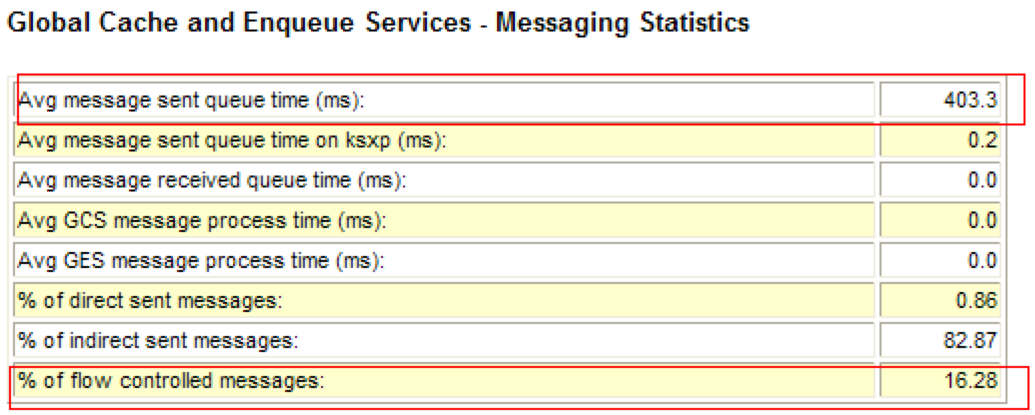
- **Avg global cache cr block receive time (ms) =**
  Avg global cache cr block build time (ms)
  + Avg global cache cr block send time (ms)
  + Avg global cache cr block flush time (ms)
  + Avg message sent queue time on ksxp(ms)
  + 其他？

87 显然不等于 0.0 + 0.0 + 3.1 + 0.2
还差了80毫秒哪去了？
这里的其他还包括什么？

通过上述分析，我们发现：
(1)节点1发送至节点2的指标正常
(2)节点2收到节点1的消息后处理指标正常
(3)节点2返回消息至节点1的指标正常

时间都消耗发送队列中的等待了！

- 但是节点1到节点2的总响应时间却是异常的
- 这说明，有1个环节肯定是异常的，就是“消息从进入队列”到“被发送出去”之间的间隔，如果RAC出现流量控制，则会导致该问题。
- 继续检查，如下所示：



Global Cache and Enqueue Services - Messaging Statistics

Avg message sent queue time (ms):	403.3
Avg message sent queue time on ksxp (ms):	0.2
Avg message received queue time (ms):	0.0
Avg GCS message process time (ms):	0.0
Avg GES message process time (ms):	0.0
% of direct sent messages:	0.86
% of indirect sent messages:	82.87
% of flow controlled messages:	16.28

比较分析

- **可以看到：**
- （1）其中Avg message sent queue time (ms) 指标就表示“消息从进入队列”到“被发送出去”之间的间隔,不过这个不是原因，该指标本身体现的是一个结果的指标。
- （2）异常时候的消息流控指标即指标“% of flow controlled messages:”达到16.28%,而正常时候该指标只有0.08！
- （3）我们通过多次不同时间点比较，均有相同结论，当流控指标较大时，就会出现问题。

什么时候出现RAC私网流量控制

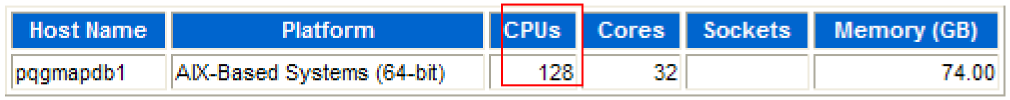
- RAC里的流控，可以简单理解为机场中经常出现的流量控制，原理相同，下面情况下可能导致流控
- (1)天气出现异常，即私网网络链路不顺畅
- (2)目标机场负载较大等，即RAC对端节点负载很高
- (3)两地服务能力不对等，即两个节点节点传输配置有差异
- 具体到我们此处的情况，私网传输指标正常，节点负载也非常低，因此，最可能的就是两端服务能力不对等。而与RAC GC传输能力不对等的也就是LMS进程了，该进程负责节点之间消息和BLOCK的传输和处理。当两个节点LMS进程个数不一致时，可能出现服务能力不对等的问题。
- 因此，继续检查两个节点的LMS进程个数。

进一步确认证据

- **节点1：**
- Wed Jul 24 16:19:22 2013
- **LMS 5**: 0 GCS shadows cancelled, 0 closed, 0 Xw survived
- **节点2：**
- Wed Jul 24 16:21:18 2013
- **LMS3** started with pid=16, OS id=17694842 at elevated priority
- **可以看到：**
- 节点1有6个LMS进程，而节点2只有4个LMS进程。
- ORACLE中，如果没有设置gcs_server_processes 参数来指定LMS进程的个数，那么数据库默认将以CPU个数来进行计算，以决定起多少个LMS进程。
- 由于检查，未发现设置该参数，因此需要继续检查CPU个数。

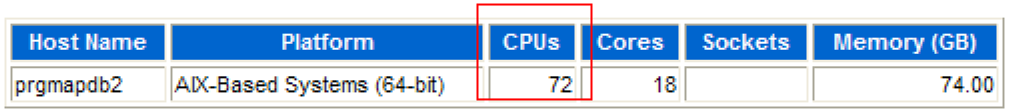
找到导致奇怪性能问题的原因

- 节点1:



Host Name	Platform	CPUs	Cores	Sockets	Memory (GB)
pqgmapdb1	AIX-Based Systems (64-bit)	128	32		74.00

- 节点2:



Host Name	Platform	CPUs	Cores	Sockets	Memory (GB)
prgmapdb2	AIX-Based Systems (64-bit)	72	18		74.00

- 可以看到,两个节点CPU个数不一样
  - 节点1逻辑CPU 128，节点2为78.

最终性能问题解决方案

- 在调整两个节点CPU个数一致并重启数据库后，两个节点LMS进程个数一致。
- 调整完成后，原来约一周一次的GC缓慢问题，不再出现。

- 其他解决方案
  - CPU不一致情况下，需设置gcs_server_processes 参数为一致

分享内容回顾与总结

中亦科技
China Etek Service

Perfecting IT service and favoring clients 'success
锻造凝练IT服务 助推用户事业发展

介绍下我们Team的其他小伙伴

黄远邦
数据库服务团队负责人

十年以上数据库维护经验，擅长人才培养、复杂故障、复杂性能问题定位和解决、无备份时的数据拯救

陈宏义

二十年以上数据库维护经验，曾就职于原厂二线支持团队,擅长复杂性能、复杂故障问题定位和解决

杨元同

十年以上数据库维护经验，曾就职于原厂售后团队，擅长复杂故障、复杂性能问题定位和解决、无备份时的数据拯救

张海亮

十年以上数据库维护经验，曾就职于原厂售后团队，擅长SQL调优、故障处理、性能调优

王远军

二十年以上数据库维护经验，曾就职于原厂售后团队，擅长故障处理、性能调优

Perfecting IT service and favoring clients 'success
锻造凝练IT服务 助推用户事业发展

介绍下我们Team的其他小伙伴

中亦科技
China Etek Service

李瑞龙

十年以上数据库维护经验，擅长复杂故障、复杂性能问题定位和解决

周永康

八年以上数据库维护经验，擅长SQL调优、故障处理、性能调优

林浩南

六年以上数据库维护经验，擅长故障处理、性能调优

佟长胜

八年以上数据库维护经验，擅长SQL调优、故障处理、性能调优

关于小y-技术即人生

中亦科技
China Etek Service

黄远邦

联系方式:
13701026113
Shadow.huang@
ce-service.com.cn
51994106@qq.com

微信号：
Shadow-huang-bj

微信号：Shadow-huang-bj

就职于北京中亦安图科技股份有限公司
中亦科技是国内领先的综合运维服务提供商,提供包括数据库、操作系统、中间件、存储、网络等全方位的运维服务

十年以上Oracle数据库维护经验，擅长数据库架构设计、复杂故障、复杂性能问题定位和解决。

数年来为数十家银行总行客户提供数据库专家服务
此外为航空、证券、基金、保险、运营商、政府、制造业等众多客户提供数据库维护服务

中亦科技
China Etek Service

THANK YOU

感谢您的关注

中亦科技吉祥物
海狸先生

Perfecting IT service and favoring clients 'success
锻造凝练IT服务 助推用户事业发展