

目 录



第 1 章 Oracle 数据库常见问题诊断方法	1
1.1 常见错误篇	1
1.1.1 ORA-12571、ORA-03113、ORA-03114、ORA-01041	1
1.1.2 ORA-01000	1
1.1.3 ORA-01545	2
1.1.4 ORA-0165x.....	2
1.1.5 ORA-01555	3
1.1.6 ORA-04031	3
1.1.7 ORA-04091	3
1.1.8 ORA-01242、ORA-01113.....	4
1.2 内部错误篇	4
1.2.1 ORA-00600 【12330】 错误.....	4
1.2.2 ORA-00604 【xxx】 错误	5
1.2.3 ORA-00600 【3339】 错误.....	5
1.2.4 ORA-00600 【13004】 错误.....	5
1.3 分布式事务篇	6
1.3.1 诊断分布式事务	6
1.3.2 检查其它节点的事务 (DBA_2PC_NEIGHBORS)	6
1.3.3 通过 DBA_2PC_PENDING 字典表检查事务的状态	6
1.3.4 检查处理结果.....	7
1.3.5 COMMIT FORCE 或 ROLLBACK FORCE 命令.....	7
1.4 OPS 或 RAC 篇	8
1.4.1 准备工作	8
1.4.2 紧急情况下的状态备份	8
1.4.3 OPS 设计、配置准则.....	9
1.4.4 OPS 常见问题.....	9
1.4.5 诊断分析步骤.....	9
1.5 非 OPS 篇	18
1.5.1 ORACLE 数据库系统常见问题：空间方面的问题	18
1.5.2 ORACLE 数据库系统常见问题：性能方面的问题	18
1.5.3 ORACLE 数据库系统常见问题：锁争用方面的问题	19
1.5.4 ORACLE 数据库系统常见问题：内存方面的问题	20
1.5.5 ORACLE 问题分析脚本	20
1.5.6 SQL*NET 篇	24
1.5.7 TNS-12154 Error 或 ORA-12154	24
1.5.8 NL-00462 Error 或 ORA-00462	25
1.5.9 NL-00405 Error 或 ORA-00405	26

1.5.10 TNS-01155 Error 或 ORA-01155	26
1.5.11 TNS-12537 、 TNS-12560、 TNS-00507 Error.....	26
1.5.12 TNS-12203 Error	27
1.5.13 TNS-12533 Error	27
1.6 备份与恢复篇	27
1.6.1 EXP-00942 或 ORA-00942、 ORA-00904 错误.....	28
1.6.2 EXP-00037 或 ORA-00037	28
1.6.3 IMP-00009 或 ORA-00009	28
1.6.4 EXP-00041 或 ORA-00041.....	29
1.6.5 IMP-00016 、 IMP-00036 、 IMP-00037 、 IMP-00038.....	29

第1章 Oracle 数据库常见问题诊断方法

1.1 常见错误篇

ORACLE 的这类错误在 ORACLE 的文档中有详细说明,但原因及措施说明不详细,本文当着重说明如何解决这类错误。

1.1.1 ORA-12571、ORA-03113、ORA-03114、ORA-01041

1. 特征

客户端(代理或应用服务器)有时报这类断连错误

2. 原因

如果偶尔出现一次,则可能为网络原因或用户异常中止,如果经常出现则为客户端与服务端的字符集不一致。

3. 措施

如果偶尔出现,可在服务端的协议配置文件 PROTOCOL.ORA 中增加一行

```
TCP.NODELAY=YES;
```

如果经常出现,则为客户端与服务端字符集不一致或网络原因。

客户端的字符集在注册表里定义:

```
HKEY__LOCAL__MACHINE/SOFTWARE/ORACLE/NLS__LANG
```

在客户端注册表中的 TCP 参数项中设置 TCPMAXDATARETRANSMISSIONS=20。

1.1.2 ORA-01000

1. 特征

达到会话允许的最大游标数

2. 原因

达到会话允许的最大游标数

3. 措施

有两种解决方法:

在初始化文件 INIT<SID>.ORA 文件中增加 OPEN_CURSORS 的数量, 一般要求大于 200。

在应用级, 与开发工具有关, 例如设置 MAXOPEN_CURSORS 等。

1.1.3 ORA-01545

1. 特征

某个回滚段不可用

2. 原因

- (1) 当使回滚段 ONLINE 时, 但回滚段不可用, 例如回滚段所在表空间 OFFLINE;
- (2) 当使回滚段 ONLINE 时, 但回滚段已 ONLINE, 例如回滚段被使用两次, 典型的案例如 OPS 方式时, 回滚段不能公有;
- (3) 删除回滚段时, 回滚段中有活动的事务;

3. 措施

- (1) 确保回滚段可用
- (2) 从 初始化文件 INIT<SID>.ORA 的参数 ROLLBACK)SEGMENTS 中删除指定的回滚段。
- (3) 可以将回滚段所在表空间删除, 取消 UNDO 事务

1.1.4 ORA-0165x

1. 特征

表空间没有足够的空间供分配

2. 原因

表空间已满; 存储参数不合理, NEXT 太小; 没有连续的区间

3. 措施

如果表空间已满, 则需为表空间增加文件; 如果存储参数不合理, 则需增加 INITIAL 和 NEXT; 如果没有连续的区间, 需要合并空闲的表空间。

查看空间碎片用 DBA_FREE_SPACE

1.1.5 ORA-01555

1. 特征

当前会话无法读到以前版本的数据

2. 原因

原因很多，主要原因有下列：回滚段太小、太少；回滚段冲突；交叉提交 (FETCH_ACROSS)

3. 措施

增加回滚段数量；

1.1.6 ORA-04031

1. 特征

共享池内存区内存不够，或产生内存碎片

2. 原因

当试图装载一个大包时或执行一个较大的存储过程时，而共享池没有连续的内存空间。

3. 措施

如果是内存不够，则增加 `SHARED_POOL_SIZE`；

如果是内存碎片，执行 `alter system flush share_pool`

1.1.7 ORA-04091

1. 特征

触发器工作不正常

2. 原因

一个行触发读取或修改变化的表（正在修改、插入）时，产生这种错误。

3. 措施

检查触发器脚本，保证引用完整性

1.1.8 ORA-01242、ORA-01113

1. 特征

介质故障导致数据库宕机

2. 原因

介质故障。

3. 措施

检查硬件故障；修改 dbshut 脚本，将其中的 STARTUP 命令修改为：

```
Startup open recover
```

```
Alter database open
```

1.2 内部错误篇

ORACLE 的错误各种各样，包括应用错误、一般错误、内部错误等，前面两类错误在 ORACLE 的文档中有说明，但内部错误没有相应的文档说明，只是请求报告 ORACLE 技术支持，本文档主要讨论 ORACLE 的内部错误，且这些内部错误在 ICD 中经常出现，仅供参考。

内部错误一般为格式为 ORA-00600 或 ORA-006XX,其中前者最普遍，后者较少见，ORA-600 中的第一个变量用于标记代码中错误的位置，第二个到第五个变量显示附加信息，例如文件号、函数号等具体信息。

1.2.1 ORA-00600 【12330】 错误

1. 特征

数据库告警日志中经常有这个错误及相应的 trace 文件

2. 原因

用户异常中断操作或客户端字符集与 SERVER 端字符集不一致

3. 措施

如果偶尔出现，则为用户异常中止，例如代理或应用服务器的断连，有时会产生这个错误；如果经常出现，则为客户端与服务端字符集不一致。

客户端的字符集在注册表里定义：

```
HKEY__LOCAL__MACHINE/SOFTWARE/ORACLE/NLS__LANG
```

1.2.2 ORA-00604 【xxx】 错误

1. 特征

在分析 SQL 语句时，查询数据字典表发生错误

2. 原因

这类错误一般与内存管理有关，有可能是由于内存泄漏导致该错误

3. 措施

如果偶尔出现，适当加大 SHARE_POOL_SIZE；如果经常出现，则需要打相应的补丁。

1.2.3 ORA-00600 【3339】 错误

1. 特征

数据冲突，包括：块格式冲突、非法索引入口

2. 原因

oracle 系统本身 bug；操作系统或介质故障

3. 措施

ORACLE 升级或打补丁；检查硬件故障

1.2.4 ORA-00600 【13004】 错误

1. 特征

逻辑冲突，例如查询返回错误的数据等

2. 原因

oracle 系统本身 bug;

3. 措施

ORACLE 升级或打补丁

1.3 分布式事务篇

对于数据库服务端到服务端的访问（如 DBLINK、复制、快照等），由于网络等原因可能会产生一个节点的事务无法恢复，与之相关的另一个节点的数据库事务挂起，因而产生分布式数据库事务问题。

1.3.1 诊断分布式事务

(1) 检查 alert<sid>.log 文件，发现相应的错误

确保网络正常，并检查 DBLINK 是” valid” 和可操作的

(2) SELECT * FROM V\$DBLINK 或 GV\$DBLINK

(3) 查找悬挂的事务（ DBA_2PC_PENDING）

```
SELECT LOCAL_TRAN_ID, GLOBAL_TRAN_ID, STATE, MIXED, HOST,
COMMIT# FROM DBA_2PC_PENDING
```

LOCAL_TRAN_ID 是本机的事务号（报告错误的机器），如果 LOCAL_TRAN_ID = GLOBAL_TRAN_ID, 即分布式事务来源于本机，也可以从本机的 alert<sid>.log 中得到 LOCAL_TRAN_ID。

1.3.2 检查其它节点的事务（DBA_2PC_NEIGHBORS）

(1) 执行下列命令：

```
SELECT LOCAL_TRAN_ID, IN_OUT, DATABASE, INTERFACE
FROM DBA_2PC_NEIGHBORS
```

(2) 在 init<sid>.ora 中检查参数 COMMIT_POINT_STRENGTH

该参数应有较大值（最好最大值）

1.3.3 通过 DBA_2PC_PENDING 字典表检查事务的状态

(1) 如果状态是 commit，则本地数据库提交成功，即不必在本数据库实施 COMMIT FORCE 或 ROLLBACK FORCE。

(2) 如果状态是 not committed(prepared), 则必需在本数据库实施 COMMIT FORCE 或 ROLLBACK FORCE, SCN 号可在 DBA_2PC_PENDING 字典表中找到。

(3) 比较个节点的 GLOBAL_TRAN_ID 及 SCN 号 (DBA_2PC_PENDING) 如果在其它节点没有这个 GLOBAL_TRAN_ID 及 SCN, 则 RECO 已经解决了这个问题, 可以不作任何事情。此时可在进程中看到 RECO 进程 (通常看不到)

(4) 如果节点存在没有提交事务

如果事务的状态是 prepared, 则必需在本数据库实施 COMMIT FORCE 或 ROLLBACK FORCE, SCN 号可在 DBA_2PC_PENDING 字典表中找到。同时, 应通过 DBA_2PC_NEIGHBORS 字典表检查有无与 DBA_2PC_PENDING 中相同的事务 ID, 如果有, 则先处理 DBA_2PC_NEIGHBORS 中的事务。

1.3.4 检查处理结果

执行了 COMMIT FORCE 或 ROLLBACK FORCE 后, 字典表 DBA_2PC_PENDING 和 DBA_2PC_NEIGHBORS 应无记录。

如果由于某种原因没有清除, 执行下列包 DBMS_TRANSACTION.purge_lost_db_entry, 这个是最好的办法, 原则上不许执行。

1.3.5 COMMIT FORCE 或 ROLLBACK FORCE 命令

(1) COMMIT FORCE 命令

如果 SCN 是 88123887 (dba_2pc_pending), 本地事务 ID 是 1.13.5197 (dba_2pc_pending 或 alert<sid>.log)

```
SVRMGR> COMMIT FORCE '本机事务 ID', '最高 SCN';
```

```
SVRMGR> COMMIT FORCE '1.13.5197', '88123887';
```

(2) ROLLBACK FORCE 命令

如果本机事务 ID 是 1.13.5197 (dba_2pc_pending 或 alert<sid>.log):

```
SVRMGR> ROLLBACK FORCE '本机事务 ID';
```

```
SVRMGR> ROLLBACK FORCE '1.13.5197';
```

(3) 强行清除事务命令

如果本机事务 ID 是 1.13.5197 (dba_2pc_pending 或 alert<sid>.log), 则:

```
SVRMGR> Execute DBMS_TRANSACTION.PURGE_LOST_DB_ENTRY  
(1.13.5197);
```

或

```
SVRMGR> Execute DBMS_TRANSACTION.PURGE_MIXED ('1.13.5197');
```

注意：只有当 RECO 无法解决问题时，采用这个命令。

1.4 OPS 或 RAC 篇

ORACLE 并行服务（即 OPS）在 ICD 中应有广泛应用，平台涉及各主流硬件平台，包括 IBM AIX、HP UX、SUN SOLARES、TRU UNIX、WINDOWS 平台，版本也较多，包括 7.3.X、8.0.X、8.1.X（8I），由于 ORACLE OPS 本身固有的特点及缺陷或使用不当，有时会产生较严重的性能问题，甚至导致数据库挂起(HUNG)。本文档就是分析、解决 OPS 的性能及 OPS 挂起问题。

由于分析、解决 OPS 问题需要对 ORACLE OPS 的工作原理有较深刻理解，所以本文档只是提供通用的分析思路及分析脚本。

该脚本主要用于 ORACLE8I 以下版本，对于 ORACLE9I 有些脚本可能不能执行。

1.4.1 准备工作

以 sys 用户运行脚本 catparr.sql，该脚本位于\$ORACLE_HOME/rdbms/admin 目录下。

1.4.2 紧急情况下的状态备份

如果数据库挂起无法执行任何命令，则只能执行状态备份命令，待重启后再进行分析。

状态备份命令如下（sys 或 internal 用户）：

```
SVRMGR> alter session set max_dump_file_size = 'UNLIMITED';
```

```
SVRMGR> alter session set events 'immediate trace name systemstate  
level 10';
```

对于 8.1.7 以上版本，执行下列命令：

```
SQL> oradebug setmypid
```

```
SQL> oradebug unlimit
```

```
SQL> oradebug setinst all
```

```
SQL> oradebug -g def dump systemstate 10
```

1.4.3 OPS 设计、配置准则

为了充分利用 OPS 的特性，应用设计及配置应严格遵循相应准则，否则，采用 OPS 系统反而导致性能及安全性下降。OPS 应用应遵循的几个准则：

- (1) 应用严格分割：例如，不同的应用分别运行在不同的实例上；
- (2) 事务分割：即一类事务分别运行于同一个实例，例如，平台的日结应与平台业务运行在一个实例上；
- (3) 功能分割：与设计相关，暂不考虑；
- (4) 表分区：大表的数据分区

ICD 产品一般采用 1、2、4 方法。

1.4.4 OPS 常见问题

- (1) OPS 挂起 (HANG)；
- (2) 性能很低，有时一台机器几乎无相应；
- (3) 分布式锁进程 LMD0 占用资源很大；

1.4.5 诊断分析步骤

1. OPS 有时挂起 (HANG)

ORACLE 数据库挂起的原因很多，也不容易分析，一般情况下，除了 ORACLE 数据库系统本身的 BUG 之外，可能应用没有按照第四章的准则配置，下面是一些检查步骤：

- (1) 检查初始化文件参数 GC_FILES_TO_LOCKS，保证该参数合理设置，gc_files_to_locks 语法：

```
GC_FILES_TO_LOCKS = "{ file_list=lock_count[!blocks][EACH]}[:]..."
```

具体设置时请参考第六章中问题 12、13、14 的分析结果，一般情况下，SYSTEM、ROLLBACK、有大量冲突目标所在的文件必须重新设置。

该参数给每一个数据文件给出一定数量的 PCM 锁，这样减少锁资源的分配开销并减少访问冲突

- (2) 减少双机之间的通讯，在初始化文件中设置：
_lm_send_direct_send=lkmgr
- (3) 检查 LM_RESS、LM_LOCKS 参数

检查 V\$resource_limit, 保证 LM_RESS、LM_LOCKS 大于该视图中的最大值, 且 lm_locks>=lm_ress

(4) DLM (分布式锁管理器) 通信量

如果 TCKT_AVAIL 值特大, 会导致系统很慢, 有时会导致 OPS 挂起; 如果 TCKT_WAIT 是 'YES', 则系统已经耗尽 DLM 资源 (没有 ticket), 因此必须确保足够的 TCKT_AVAIL。

```
select * from gv$dml_traffic_controller
order by TCKT_AVAIL;
```

2. 性能很低

(1) 检查 DLM 资源

检查 gv\$dml_traffic_controller 视图, 如果 TCKT_AVAIL 值特大, 会导致系统很慢

```
select * from gv$dml_traffic_controller
order by TCKT_AVAIL;
```

3. 分布式锁进程 LMD0 占用资源很大

该问题一般是 ORACLE 的 BUG, 需要打补丁。另外, 也可通过下列设置减少 LMD0 的工作。

```
_lm_send_direct_send=lkmgr
```

4. OPS 问题通用分析方法

所有 OPS 问题均可通过下列脚本进行分析, 这些脚本检查 OPS 涉及的各方面的内容, 所以, 对所有 OPS 问题, 如果时间允许, 应按下列方法全面检查。另外, 这些脚本也可用于日常维护。

(1) 等待会话

下列 SQL 语句查询运行时间较长, 且等待一个非空闲的等待事件, 事件名可从 V\$EVENT_NAME 中查找。

```
select sw.inst_id, sw.sid, sw.state, sw.event, sw.seconds_in_wait
seconds,
sw.p1, sw.p2, sw.p3, sa.sql_text last_sql
from gv$session_wait sw, gv$session s, gv$sqlarea sa
where sw.event not in
('rdbms ipc message','smon timer','pmon timer',
'SQL*Net message from client','lock manager wait for remote message',
'ges remote message', 'client message', 'pipe get', 'Null event', 'PX
```

```

Idle Wait',
'single-task message', 'PX Deq: Execution Msg', 'KXFQ: kxfqdeq - normal
dequeue',
'listen endpoint status','slave wait','wakeup time manager')
and seconds_in_wait > 0
and (sw.inst_id = s.inst_id and sw.sid = s.sid)
and (s.inst_id = sa.inst_id and s.sql_address = sa.address)
order by seconds desc;

```

其中 p1、p2、p3 通过下列语句查找

```

select distinct event, p1text, p2text, p3text
from gv$session_wait sw
where sw.event not in ('rdbms ipc message','smon timer','pmon timer',
'SQL*Net message from client','lock manager wait for remote message',
'ges remote message', 'client message', 'SQL*Net more data from client',
'pipe get', 'Null event', 'PX Idle Wait', 'single-task message',
'wakeup time manager')
and seconds_in_wait > 0
order by event;

```

(2) PCM 锁资源阻挡

这个脚本显示正在持有锁的会话正在阻挡另一个会话，其中 **grant_level** 表示被授予的 PCM 锁，**request_level** 表示正在请求的锁，**lockstate** 表示锁的状态，**seconds_in_wait** 表示会话等待时间。

```

select dl.inst_id, s.sid, p.spid, dl.resource_name1,
decode(substr(dl.grant_level,1,8),'KJUSERNL','Null','KJUSERCR','Row-
S (SS)',
'KJUSERCW','Row-X (SX)','KJUSERPR','Share','KJUSERPW','S/Row-X
(SSX)',
'KJUSEREX','Exclusive',request_level) as grant_level,
decode(substr(dl.request_level,1,8),'KJUSERNL','Null','KJUSERCR','Ro
w-S (SS)',
'KJUSERCW','Row-X (SX)','KJUSERPR','Share','KJUSERPW','S/Row-X
(SSX)',
'KJUSEREX','Exclusive',request_level) as request_level,
decode(substr(dl.lockstate,1,8),'KJUSERGR','Granted','KJUSEROP','Ope
ning',
'KJUSERCA','Canceling','KJUSERCV','Converting') as lockstate,
s.sid, sw.event, sw.seconds_in_wait sec
from gv$dml_locks dl, gv$process p, gv$session s, gv$session_wait sw
where blocker = 1
and (dl.inst_id = p.inst_id and dl.pid = p.spid)
and (p.inst_id = s.inst_id and p.addr = s.paddr)

```

```
and (s.inst_id = sw.inst_id and s.sid = sw.sid)
order by sw.seconds_in_wait desc;
```

(3) PCM 锁等待

这个脚本显示正在等待锁资源的会话，其中 **grant_level** 表示被授予的 PCM 锁，**request_level** 表示正在请求的锁，**lockstate** 表示锁的状态，**seconds_in_wait** 表示会话等待时间。

```
select dl.inst_id, s.sid, p.spid, dl.resource_name1,
decode(substr(dl.grant_level,1,8),'KJUSERNL','Null','KJUSERCR','Row-
S (SS)',
'KJUSERCW','Row-X          (SX)','KJUSERPR','Share','KJUSERPW','S/Row-X
(SSX)',
'KJUSEREX','Exclusive',request_level) as grant_level,
decode(substr(dl.request_level,1,8),'KJUSERNL','Null','KJUSERCR','Ro
w-S (SS)',
'KJUSERCW','Row-X          (SX)','KJUSERPR','Share','KJUSERPW','S/Row-X
(SSX)',
'KJUSEREX','Exclusive',request_level) as request_level,
decode(substr(dl.lockstate,1,8),'KJUSERGR','Granted','KJUSEROP','Ope
ning',
'KJUSERCA','Cancelling','KJUSERCV','Converting') as lockstate,
s.sid, sw.event, sw.seconds_in_wait sec
from gv$dml_locks dl, gv$process p, gv$session s, gv$session_wait sw
where blocked = 1
and (dl.inst_id = p.inst_id and dl.pid = p.spid)
and (p.inst_id = s.inst_id and p.addr = s.paddr)
and (s.inst_id = sw.inst_id and s.sid = sw.sid)
order by sw.seconds_in_wait desc;
```

(4) 本地锁等待

这个脚本显示本地锁等待，其中 **addr** 表示锁地址，**type** 表示锁类型，如果锁类型是 **TM**，则 **id1** 表示目标 ID，**id2** 为 0；如果锁类型是 **TX**，则 **id1** 表示事务槽数（可不管），

```
select l.inst_id, l.sid, l.addr, l.type, l.id1, l.id2,
decode(l.block,0,'blocked',1,'blocking',2,'global') block,
sw.event, sw.seconds_in_wait sec
from gv$lock l, gv$session_wait sw
where (l.sid = sw.sid and l.inst_id = sw.inst_id)
and l.block in (0,1)
order by l.type, l.inst_id, l.sid;
```

(5) 门闩锁保持

该锁是 oracle 的内部锁，用户无法控制，一般持有时间很短，可不考虑。但如果发现门闩锁冲突或'latch free'等待时间，则应执行下列脚本检查。

```
select distinct lh.inst_id, s.sid, s.username, p.username os_user,
lh.name
from gv$latchholder lh, gv$session s, gv$process p
where (lh.sid = s.sid and lh.inst_id = s.inst_id)
and (s.inst_id = p.inst_id and s.paddr = p.addr)
order by lh.inst_id, s.sid;
```

门闩锁的命中率脚本如下：

```
select distinct lh.inst_id, s.sid, s.username, p.username os_user,
lh.name
from gv$latchholder lh, gv$session s, gv$process p
where (lh.sid = s.sid and lh.inst_id = s.inst_id)
and (s.inst_id = p.inst_id and s.paddr = p.addr)
order by lh.inst_id, s.sid;
```

或

```
select inst_id, name latch_name,
round((immediate_gets/(immediate_gets+immediate_misses)), 3)
hit_ratio,
round(sleeps/decode(immediate_misses,0,1,immediate_misses),3)
"SLEEPS/MISS"
from gv$latch
where round((immediate_gets/(immediate_gets+immediate_misses)), 3)
< .99
and immediate_gets + immediate_misses > 0
order by round((immediate_gets/(immediate_gets+immediate_misses)), 3);
```

(6) 全局缓存一致性读性能脚本

这个脚本显示了平均一致性读的等待时间（AVG CR BLOCK RECEIVE TIME），一般情况下该值为 15 毫秒左右，该值与 DB_MULTI_BLOCK_READ_COUNT 参数有关，DB_MULTI_BLOCK_READ_COUNT 值越大，等待时间也越大。

```
select b1.inst_id, b2.value "GCS CR BLOCKS RECEIVED",
b1.value "GCS CR BLOCK RECEIVE TIME",
((b1.value / b2.value) * 10) "AVG CR BLOCK RECEIVE TIME (ms)"
from gv$sysstat b1, gv$sysstat b2
where b1.name = 'global cache cr block receive time' and
b2.name = 'global cache cr blocks received' and b1.inst_id = b2.inst_id ;
```

(7) 全局缓存锁性能

该脚本显示平均全局队列得到时间 (AVG GLOBAL LOCK GET TIME)，一般情况下，

AVG GLOBAL LOCK GET TIME 为 20-30 毫秒 (包括分配全局队列和初始化时间)。如果该值很大，则系统必然存在许多超时时间，请按上述方法检查。

```
select b1.inst_id, (b1.value + b2.value) "GLOBAL LOCK GETS",
b3.value "GLOBAL LOCK GET TIME",
(b3.value / (b1.value + b2.value) * 10) "AVG GLOBAL LOCK GET TIME (ms)"
from gv$sysstat b1, gv$sysstat b2, gv$sysstat b3
where b1.name = 'global lock sync gets' and
b2.name = 'global lock async gets' and b3.name = 'global lock get time'
and b1.inst_id = b2.inst_id and b2.inst_id = b3.inst_id;
```

(8) 资源应用

这个脚本检查资源利用情况。

```
select inst_id, resource_name, current_utilization, max_utilization,
initial_allocation
from gv$resource_limit
where max_utilization > 0
order by inst_id, resource_name;
```

(9) DLM (分布式锁管理器) 通信量

如果该值特大，会导致系统很慢，有时会导致 OPS 挂起。如果 TCKT_WAIT 是 'YES'，则系统已经耗尽 DLM 资源 (没有 ticket)，因此必须确保足够的 TCKT_AVAIL。

```
select * from gv$dml_traffic_controller
order by TCKT_AVAIL;
```

(10) DLM 杂项

```
select * from gv$dml_misc;
```

(11) 锁转换详细信息

```
select * from gv$lock_activity;
```

(12) 前 10 个写 PING/FUSION 数最大的的目标

```
select inst_id, name, kind, file#, status, BLOCKS,
READ_PINGS, WRITE_PINGS
from (select p.inst_id, p.name, p.kind, p.file#, p.status,
count(p.block#) BLOCKS, sum(p.forced_reads) READ_PINGS,
sum(p.forced_writes) WRITE_PINGS
from gv$ping p, gv$datafile df
where p.file# = df.file# (+)
group by p.inst_id, p.name, p.kind, p.file#, p.status
order by sum(p.forced_writes) desc)
where rownum < 11
```



```
order by WRITE_PINGS desc;
```

(13) 前 10 个读 PINGING/FUSION 数最大的目标

```
select inst_id, name, kind, file#, status, BLOCKS,
READ_PINGS, WRITE_PINGS
from (select p.inst_id, p.name, p.kind, p.file#, p.status,
count(p.block#) BLOCKS, sum(p.forced_reads) READ_PINGS,
sum(p.forced_writes) WRITE_PINGS
from gv$ping p, gv$datafile df
where p.file# = df.file# (+)
group by p.inst_id, p.name, p.kind, p.file#, p.status
order by sum(p.forced_reads) desc)
where rownum < 11
order by READ_PINGS desc;
```

(14) 前 10 个 FALSE PINGING 数最大的目标

这个可通过 `gc_files_to_locks` 避免。

```
select inst_id, name, kind, file#, status, BLOCKS,
READ_PINGS, WRITE_PINGS
from (select p.inst_id, p.name, p.kind, p.file#, p.status,
count(p.block#) BLOCKS, sum(p.forced_reads) READ_PINGS,
sum(p.forced_writes) WRITE_PINGS
from gv>false_ping p, gv$datafile df
where p.file# = df.file# (+)
group by p.inst_id, p.name, p.kind, p.file#, p.status
order by sum(p.forced_writes) desc)
where rownum < 11
order by WRITE_PINGS desc;
```

(15) 前 10 个等待事件

```
select inst_id, event, time_waited, total_waits, total_timeouts
from (select inst_id, event, time_waited, total_waits, total_timeouts
from gv$system_event where event not in ('rdbms ipc message','smon
timer',
'pmon timer','SQL*Net message from client',
'lock manager wait for remote message','ges remote message', 'client
message',
'SQL*Net more data from client', 'pipe get', 'Null event', 'PX Idle Wait',
'single-task message', 'wakeup time manager')
order by time_waited desc)
where rownum < 11
order by time_waited desc;
```

(16) 显示会话、进程、程序

```
select p.inst_id, s.sid, s.serial#, p.pid, p.spid, p.program,
```

```

s.username,
p.username os_user, sw.event, sw.seconds_in_wait sec
from gv$process p, gv$session s, gv$session_wait sw
where (p.inst_id = s.inst_id and p.addr = s.paddr)
and (s.inst_id = sw.inst_id and s.sid = sw.sid)
order by p.inst_id, s.sid;

```

(17) 没有授予等待会话的 PCM 锁

这个脚本显示当前没有授予任何等待会话的锁，由于冲突这些锁很可能有问题。

```

select dl.inst_id, s.sid, p.spid, dl.resource_name1,
decode(substr(dl.grant_level,1,8),'KJUSERNL','Null','KJUSERCR','Row-
S (SS)',
'KJUSERCW','Row-X          (SX)','KJUSERPR','Share','KJUSERPW','S/Row-X
(SSX)',
'KJUSEREX','Exclusive',request_level) as grant_level,
decode(substr(dl.request_level,1,8),'KJUSERNL','Null','KJUSERCR','Ro
w-S (SS)',
'KJUSERCW','Row-X          (SX)','KJUSERPR','Share','KJUSERPW','S/Row-X
(SSX)',
'KJUSEREX','Exclusive',request_level) as request_level,
decode(substr(dl.lockstate,1,8),'KJUSERGR','Granted','KJUSEROP','Ope
ning',
'KJUSERCA','Cancelling','KJUSERCV','Converting') as lockstate,
s.sid, sw.event, sw.seconds_in_wait sec
from gv$dml_locks dl, gv$process p, gv$session s, gv$session_wait sw
where lockstate not like 'KJUSERGRANTED%'
and (dl.inst_id = p.inst_id and dl.pid = p.spid)
and (p.inst_id = s.inst_id and p.addr = s.paddr)
and (s.inst_id = sw.inst_id and s.sid = sw.sid)
and sw.event not in ('rdbms ipc message','smon timer','pmon timer',
'SQL*Net message from client','lock manager wait for remote message',
'ges remote message', 'client message', 'pipe get', 'Null event', 'PX
Idle Wait',
'single-task message', 'PX Deq: Execution Msg', 'KXFQ: kxfqdeq - normal
dequeue',
'listen endpoint status','slave wait','wakeup time manager')
and sw.seconds_in_wait > 0
order by sec desc;

```

PCM 锁没有授予等待会话的锁全部信息:

```

select * from gv$dml_all_locks
where resource_name1 in

```

```
(select dl.resource_name1
from gv$dml_locks dl, gv$process p, gv$session s, gv$session_wait sw
where lockstate not like 'KJUSERGRANTED%'
and (dl.inst_id = p.inst_id and dl.pid = p.spid)
and (p.inst_id = s.inst_id and p.addr = s.paddr)
and (s.inst_id = sw.inst_id and s.sid = sw.sid)
and sw.event not in ('rdbms ipc message','smon timer','pmon timer',
'SQL*Net message from client','lock manager wait for remote message',
'ges remote message', 'client message', 'SQL*Net more data from client',
'pipe get', 'Null event', 'PX Idle Wait', 'single-task message',
'wakeup time manager') and sw.seconds_in_wait > 0)
order by resource_name1, inst_id, lockstate, pid;
```

PCM 锁没有授予等待会话的资源全部信息:

```
select * from gv$dml_ress
where resource_name in (select dl.resource_name1
from gv$dml_locks dl, gv$process p, gv$session s, gv$session_wait sw
where lockstate not like 'KJUSERGRANTED%'
and (dl.inst_id = p.inst_id and dl.pid = p.spid)
and (p.inst_id = s.inst_id and p.addr = s.paddr)
and (s.inst_id = sw.inst_id and s.sid = sw.sid)
and sw.event not in ('rdbms ipc message','smon timer','pmon timer',
'SQL*Net message from client','lock manager wait for remote message',
'ges remote message', 'client message', 'pipe get', 'Null event', 'PX
Idle Wait',
'single-task message', 'PX Deq: Execution Msg', 'KXFQ: kxfqdeq - normal
dequeue',
'listen endpoint status','slave wait','wakeup time manager')
and sw.seconds_in_wait > 0)
order by resource_name, inst_id, master_node;
```

(18) 当前等待的会话

```
select sw.inst_id, sw.sid, sw.seconds_in_wait sec, sa.sql_text sql
from gv$session_wait sw, gv$session s, gv$sqlarea sa
where sw.sid = s.sid (+)
and sw.inst_id = s.inst_id (+)
and s.sql_address = sa.address
and sw.event not in ('rdbms ipc message','smon timer','pmon timer',
'SQL*Net message from client','lock manager wait for remote message',
'ges remote message', 'client message', 'pipe get', 'Null event', 'PX
Idle Wait',
'single-task message', 'PX Deq: Execution Msg', 'KXFQ: kxfqdeq - normal
dequeue',
'listen endpoint status','slave wait','wakeup time manager')
```

```
and seconds_in_wait > 0  
order by sw.seconds_in_wait desc;
```

1.5 非 OPS 篇

1.5.1 ORACLE 数据库系统常见问题：空间方面问题

1. 现象

随着数据库使用时间的增长，数据库系统存储的数据就越多，不断增长的数据可能导致空间不足、目标的范围分配数太大、SQL 语句的性能下降等问题，因此应该经常检查空间使用情况。

2. 解决思路

定期检查主要表空间的使用情况，执行表空间的剩余空间检查脚本（语句 4），剩余空间应该保持在 20% 以上，否则需要备份、增加数据文件或清理历史数据；

注意：空间的标志位。有时空间高位标志位远远大于实际使用的空间数量，此时应截断高位标志位。

表或索引的 Extents 数（语句 1）应该小于 50，核心表数据表 Extents 数应该小于 100，否则要参考集成案例来调整表的存储参数。由于索引对 EXTENT 比较敏感，所以，对于有大量 UPDATE、INSERT 操作的索引，EXTENT 应更小。

空间碎片问题：对于 ICD 业务来讲，由于 DDL 操作较少，所以，表空间碎片问题基本不存在，但表、索引的空间碎片问题普遍存在，例如 COMMONINFOMATION 表及其索引。此时需要对表数据备份、TRUNCATE、导入操作，对索引需要 REBUILD。

1.5.2 ORACLE 数据库系统常见问题：性能方面问题

1. 现象

座席端的调用慢甚至死机，应用服务器队列全忙、不断重连并最终断连。在数据库服务器上的现象是 CPU 资源或者 IO 资源消耗很大。

2. 解决思路

ORACLE 需要设置的参数较多,尤其是关于 SGA 的尺寸的设置对性能影响较大,当出现以上性能问题时,首先应该按照《ORACLE 数据库安装配置检查要求》逐项检查,调整不合理的参数。

操作系统资源检查:包括 CPU、IO、内存、队列等,具体检查方法请参照小型机日常维护检查手册。

在确认参数设置及系统资源没有问题后就需要查找应用本身的问题,由于数据库中数据的数量、分布的变化,可能导致 SQL 语句的性能变得较差,进而导致性能问题出现。这种情况需要找出性能较差的语句并进行分析、优化。性能较差的语句一般分为下列几种情况:

物理读比较大,这种情况很常见,在操作系统命令行下执行 `vmstat 2 10`, 可以看到 `pi,po` 项数值大于 0,表示内存中有换入换出,执行语句 12 查找物理读最多的前 20 条语句,并查看这些语句的查询计划(PL/SQL DEVELOP 工具中将 SQL 语句拷贝到 SQL Window 中按 F5),分析是否可优化。

逻辑读比较大,这种情况下一般 CPU 占用较高,可以执行语句 13 查找逻辑读最多的前 20 条语句,并查看这些语句的查询计划(方法同上),分析是否可优化。

如现场无法优化可将上述语句涉及的表、存储过程的脚本发给优化部姚理或者相关人员处理。

没有合适的索引或没有用到合适索引,需要用上述方法分析

应用设计存在缺陷,需要用上述方法分析

1.5.3 ORACLE 数据库系统常见问题:锁争用方面问题

1. 现象

这种情况下的现象也是座席端的调用慢甚至死机,应用服务器队列全忙、不断重连并最终断连。但与上节中性能问题不同的是,在数据库服务器上的现象是 CPU 资源或者 IO 资源消耗并不大。

2. 解决思路

当系统中有会话修改数据后长时间不提交事务或者修改数据的 SQL 语句本身运行时间就较长时,该会话会保持所修改数据的锁,并阻塞其他试图修改这部分数据的会话,有时甚至会话之间相互阻塞,导致死锁出现。

可以运行语句 9 查找出所有的锁，正常情况下应该没有锁或者锁的持续时间很短，如果锁的持续时间较长（大于 10 秒）则可能有问题。

执行语句 8 查看持有该锁的会话 ID 的等待事件。

执行语句 10 查看持有该锁的会话 ID 当前正在执行的 SQL 语句。

将以上执行语句的结果以及相关 SQL 语句涉及的表、存储过程的脚本发给优化部姚理或者相关人员处理。如果某会话持有锁时间太长且阻塞其他会话，可以使用语句 14 杀死该会话。

1.5.4 ORACLE 数据库系统常见问题：内存方面问题

1. 现象

内存方面问题目前有两种：1、ORACLE 本身内存泄露；2、SGA 中剩余内存不足，生成大量 TRACE 文件，严重占用 IO 资源。

2. 解决思路

为尽早发现内存泄露问题，应该定期检查小型机上剩余内存，周期为一周或者半个月，方法为使用 vmstat 命令查看 memory free 字段，单位为页（每页大小一般为 4K），如果剩余内存持续减少则存在内存泄露，请报告优化部姚理或者相关人员处理。

执行语句 15 和语句 16 查看 SGA 的信息，应该只有 NAME 字段为 db_block_buffers 或 sql area 的记录内存字节数（BYTES 字段）较大，其他项的大小应该不到 10%，如果有其他项占用异常的大请将上述语句执行结果报告优化部姚理或者相关人员处理。

1.5.5 ORACLE 问题分析脚本

查看当前用户下指定表已分配的空间以及扩展次数，例如查看表 SERVICEINFO

```
SELECT SUBSTRB(SEGMENT_NAME, 1, 30) "SEGMENT_NAME",  
SUM(BYTES) / 1024 / 1024 "Alloc(MB)",  
COUNT(*) "Extents"  
FROM USER_EXTENTS  
WHERE SEGMENT_NAME = 'SERVICEINFO'  
GROUP BY SEGMENT_NAME;
```

查看当前用户下占用空间最多的对象

```
SELECT SUBSTRB(SEGMENT_NAME, 1, 30) "SEGMENT_NAME",
```

```

SUM(Bytes) / 1024 / 1024 "Alloc(MB)",
COUNT(*) "Extents"
FROM USER_EXTENTS
GROUP BY SEGMENT_NAME
HAVING SUM(Bytes) / 1024 / 1024 >= 10
ORDER BY 2 DESC;

```

查看当前用户下区间扩展最多的对象

```

SELECT SUBSTRB(SEGMENT_NAME, 1, 30) "SEGMENT_NAME",
SUM(Bytes) / 1024 / 1024 "Alloc(MB)",
COUNT(*) "Extents"
FROM USER_EXTENTS
GROUP BY SEGMENT_NAME
HAVING COUNT(*) >= 10
ORDER BY 3 DESC;

```

查看系统所有表空间的空间使用情况

```

SELECT D.TABLESPACE_NAME "Name", D.STATUS "Status",
TO_CHAR((A.BYTES / 1024 / 1024), '99,990.9') "Size (M)",
TO_CHAR(((A.BYTES - DECODE(F.BYTES, NULL, 0, F.BYTES)) / 1024 /
1024), '99,990.9') "Used (M)",
TO_CHAR((DECODE(F.BYTES, NULL, 0, F.BYTES) / 1024 / 1024), '99,990.9')
"Free (M)"
FROM SYS.DBA_TABLESPACES D, SYS.SM$TS_AVAIL A, SYS.SM$TS_FREE F
WHERE D.TABLESPACE_NAME = A.TABLESPACE_NAME
AND F.TABLESPACE_NAME (+) = D.TABLESPACE_NAME;

```

查看所有表空间的数据文件使用情况

```

SELECT d.file_name "Name", d.tablespace_name "Tablespace", v.status
"Status",
TO_CHAR((d.bytes / 1024 / 1024), '99999990.000') "Size (M)",
NVL(TO_CHAR(((d.bytes - SUM(s.bytes)) / 1024 / 1024), '99999990.000'),
TO_CHAR((d.bytes / 1024 / 1024), '99999990.000')) "Used (M)"
FROM sys.dba_data_files d, sys.dba_free_space s, sys.v_$datafile v
WHERE (s.file_id (+)= d.file_id)
AND (d.file_name = v.name)
GROUP BY d.file_name, d.tablespace_name, v.status, d.bytes;

```

查看表空间碎片情况, 以下语句得到的结果中 **FSFI** 字段小于 **30%** 的表空间为碎片比较严重的表空间

```
SELECT TABLESPACE_NAME, SQRT(MAX(BLOCKS) / SUM(BLOCKS))
(100 / SQRT(SQRT(COUNT(BLOCKS)))) FSFI
FROM DBA_FREE_SPACE
GROUP BY TABLESPACE_NAME
ORDER BY 1;
```

查看所有会话的信息，包括会话 ID、用户名、状态、客户端机器名、运行的程序以及登陆时间

```
SELECT
sid,serial#,username,status,machine,terminal,program,logon_time FROM
V$SESSION;
```

查看所有会话的等待信息

```
SELECT sid,event,seconds_in_wait,state FROM v$session_wait;
```

查看所有会话的锁

```
SELECT sid,TYPE,id1,id2,lmode,request,ctime,block FROM v$lock WHERE
TYPE IN ('TM','TX')
```

一般情况下返回记录数应为 0，如果有记录应该检查等待时间（字段 **ctime**，单位秒），如果等待时间较长则需要检查该会话是否被其他会话的锁阻塞，字段 **request** 不为 0 表示该会话在等待其他会话释放锁而被阻塞，字段 **request** 为 0 表示该会话已获得锁，**block** 不为 0 表示该会话阻塞了其他会话。

查看指定会话正在执行的 SQL 语句

```
SELECT SQL_Text FROM V$SQLTEXT A, V$SESSION B
WHERE A.Hash_Value = B.SQL_Hash_Value
AND B.SID = 29
ORDER BY Piece;
```

查看被锁住的对象

```
SELECT b.OWNER "Owner",b.OBJECT_NAME "Loked Object",a.SESSION_ID
"Session ID",a.ORACLE_USERNAME "Oracle User",a.OS_USER_NAME "OS
User",a.PROCESS "OS Process ID",a.LOCKED_MODE "Locked Mode" FROM
v$locked_object a,dba_objects b WHERE a.OBJECT_ID = b.OBJECT_ID
```

查看 SQL 缓冲池中物理读最多的语句，一般平均读硬盘数大于 10000 则性能很差

```
SELECT SQL_Text, disk_reads/executions "Avg Disk_Reads", executions
```



```
FROM V$SQLArea WHERE executions > 0 ORDER BY disk_reads/executions DESC;
```

查看 SQL 缓冲池中逻辑读最多的语句，一般平均读内存数大于 10000 则性能很差

```
SELECT SQL_Text, buffer_gets/executions "Avg Disk_Reads", executions
FROM V$SQLArea t WHERE executions > 0 ORDER BY buffer_gets/executions
DESC;
```

杀死指定的会话

```
ALTER SYSTEM KILL SESSION '<SID>, <SERIAL#>';
```

其中字段 SID 和 SERIAL# 的值可以从 V\$SESSION 中得到（见上一条语句）。

注意：使用杀死会话语句前先运行以下语句得到要杀死语句的操作系统进程 ID，因为有可能会话已经僵死，杀会话的语句无法杀死该会话，只能将其状态标记为 'KILLED'，这时需要在操作系统的命令行下使用 KILL 或 KILL -9 根据得到的操作系统进程 ID 来杀死操作系统进程。

```
SELECT B.spid FROM V$SESSION A, V$PROCESS B WHERE A.PAddr = B.Addr AND
A.SID = <SID>;
```

查看 SGA 大小

```
SELECT * FROM v$sga
```

查看 SGA 详细信息，注意：应该只有 NAME 字段为 db_block_buffers 或 sql area 的记录内存字节数（BYTES 字段）较大，其他项应该比较小，此外 free memory 应该大于 20M

```
SELECT * FROM v$sgastat
```

查看定时任务

```
SELECT
job,log_user,last_date,this_date,next_date,total_time,broken,INTERVA
L,what FROM DBA_JOBS
```

查看正在运行的定时任务

```
SELECT * FROM dba_jobs_running
```

1.5.6 SQL*NET 篇

1.5.7 TNS-12154 Error 或 ORA-12154

1. 特征

SQL*NET 没有找到连接串或别名

2. 原因 1

(1) 没有找到 TNSNAMES.ORA 文件，该文件的缺省路径为：

- Windows 95/98 client

SQL*Net 2.x - ORAWIN95\NETWORK\ADMIN

Net8 - ORAWIN95\NET80\admin

Net8i - ORACLE\ORA81\NETWORK\ADMIN

- Windows NT client

SQL*Net 2.x - ORANT\NETWORK\ADMIN

Net8 - ORANT\NET80\ADMIN

Net8i - ORACLE\ORA81\NETWORK\ADMIN

- UNIX Client

\$ORACLE_HOME/NETWORK/ADMIN

or /etc

or /var/opt/oracle

- 在客户端设置 TNS_ADMIN

在注册表 HKEY_LOCAL_MACHINE\Software\Oracle 设置 TNS_ADMIN.

TNS_ADMIN 指向 TNSNAMES.ORA 文件的位置

3. 措施

按上述说明检查文件位置

4. 原因 2

TNSNAMES.ORA 文件中内容格式不对

5. 措施

检查文件格式，标准格式如下：

```
DEV1.WORLD =  
    (DESCRIPTION = (ADDRESS_LIST =  
(ADDRESS = (PROTOCOL = TCP) (Host = 145.45.78.56) (Port = 1521) ) )  
    (CONNECT_DATA = (SID = ORCL) ) )
```

6. 原因 3

TNSNAMES.ORA 与 SQLNET.ORA 不一致

7. 措施

SQLNET.ORA 格式如下:

```
TRACE_LEVEL_CLIENT = OFF  
SQLNET.AUTHENTICATION_SERVICES = (NONE)  
NAMES.DIRECTORY_PATH = (TNSNAMES)  
AUTOMATIC_IPC = OFF
```

因为没有 NAMES.DEFAULT_DOMAIN=world, 所以 DEV1.WORLD 连接串不能连接, 可以在 TNSNAMES.ORA 中增加

```
DEV1 =  
    (DESCRIPTION = (ADDRESS_LIST =  
(ADDRESS = (PROTOCOL = TCP) (Host = 145.45.78.56) (Port = 1521) ) )  
    (CONNECT_DATA = (SID = ORCL) ) )
```

此时可连接。

1.5.8 NL-00462 Error 或 ORA-00462

1. 特征

监听器无法启动

2. 原因

LISTENER.ORA 文件内容错误, 例如, 括号不匹配、参数名错误等、多余的空格等。

3. 措施

重新编辑文件

1.5.9 NL-00405 Error 或 ORA-00405

1. 特征

LISTENER.ORA 文件不可读或找不到

2. 原因

LISTENER.ORA 文件内容错误，例如，括号不匹配、参数名错误等、多余的空格等。

3. 措施

确保文件位置在缺省目录下 \$ORACLE_HOME/network/admin，或设置环境变量 TNS_ADMIN 指向这个文件所在位置。

1.5.10 TNS-01155 Error 或 ORA-01155

1. 特征

LISTENER.ORA 文件中 SID_LIST_LISTENER 变量赋值不对

2. 原因

LISTENER.ORA 文件内容错误，SID_LIST_LISTENER 变量赋值不对。

3. 措施

确保文件中赋值正确，\$ORACLE_HOME/network/admin/samples 目录下有样例文件，可供参考。

1.5.11 TNS-12537、TNS-12560、TNS-00507 Error

1. 特征

LISTENER.ORA 文件中引用了一个无效的协议适配程序

2. 原因

LISTENER.ORA 文件中 PROCOTOL 变量赋值不对。

3. 措施

确保文件中赋值正确，一般用 TCP 协议，`$ORACLE_HOME/network/admin/samples` 目录下有样例文件，可供参考。用 `adapters tnslnr` 检查协议程序

1.5.12 TNS-12203 Error

1. 特征

不能与目标连接

2. 原因

监听器是否启动；别名是否配置

3. 措施

检查监听器是否启动，例如 `lsnrctl status`；检查 `TNSNAMES.ORA` 的别名配置

1.5.13 TNS-12533 Error

1. 特征

不能与目标连接

2. 原因

非法的 `address` 参数，

3. 措施

检查 `TNSNAMES.ORA` 的 `ADDRESS` 参数

1.6 备份与恢复篇

如果采用非归档方式，所以日常用的备份及恢复方式是 `EXPORT` 或 `IMPORT` 工具，下列说明也只针对这种备份与恢复方式。

1.6.1 EXP-00942 或 ORA-00942、ORA-00904 错误

1. 特征

装载的表或视图不存在

2. 原因

EXPORT 或 IMPORT 所需的表或视图不存在

3. 措施

以 sys 用户执行 catexp.sql 脚本, 该脚本位于 \$ORACLE_HOME/rdbms/admin

1.6.2 EXP-00037 或 ORA-00037

1. 特征

EXPORT/IMPORT 的兼容性问题

2. 原因

服务端、客户端的版本不一致

3. 措施

应遵循下列准则:

- 服务端、客户端版本一致, 基本兼容, 正常
- 使用 x 版本倒出, x 版本导入, 基本兼容, 正常
- 使用 x 版本倒出, y 版本导入, 且 $y > x$, 向上兼容, 正常
- 使用 y 版本倒出, x 版本导入, 且 $y > x$, 向下兼容, 正常
- 使用 x 版本 export 从数据库 y 倒出, x 版本导入数据库 x 或 y, 且 $y > x$, 交叉兼容, 不宜保证, 但有时可行, 可通过修改 catexp.sql 脚本保证

1.6.3 IMP-00009 或 ORA-00009

1. 特征

倒出文件出现错误

2. 原因

导入时, 无法找到文件头

3. 措施

重新倒出；传输文件时，采用二进制模式

1.6.4 EXP-00041 或 ORA-00041

1. 特征

字符集问题

2. 原因

服务端、客户端的字符集不一致

3. 措施

应遵循下列准则：

若数据库字符集是 **a**，**exp** 回话字符集是 **b**，则倒出文件中的数据字符集是 **b**；若 **imp** 回话的字符集是 **c**，目的数据库是字符集 **d**，则 **b** 到 **c** 的转换是由 **imp** 完成，但字符集 **b** 与 **c** 的比率必须是 1。设计字符集转换时，需要注意可能丢失某些信息，若被转换的字符集大于源字符集（即是超集），则不会丢失信息。

1.6.5 IMP-00016、IMP-00036、IMP-00037、IMP-00038

1. 特征

不能完成从字符集 **b** 到 **c** 的转换

2. 原因

字符集不一致

3. 措施

将 **imp** 客户端的字符集设为 **b**，字符集在 **props\$** 核心表中。