

目录

[文档修改记录 2](#)

[1. 前言： 5](#)

[2. 简单命令使用 5](#)

[2.1 进入 SQL*Plus 5](#)

[2.2 退出 SQL*Plus 5](#)

[2.3 在 sqlplus 下得到帮助信息 6](#)

[2.4 显示表结构命令 DESCRIBE 6](#)

[2.5 SQL*Plus 中的编辑命令 6](#)

[2.6 调用外部系统编辑器 6](#)

[2.7 运行命令文件 7](#)

[2.8 关于侦听 7](#)

[3. ORACLE 的启动和关闭 8](#)

[3.1 在单机环境下 8](#)

[3.2 在双机环境下 9](#)

[4. 数据库管理员日常工作 9](#)

[4.1 检查 alterSID.log 9](#)

[4.2 环境确认 10](#)

[5. 数据库日常操作 SQL 11](#)

[5.1 查看表空间物理文件的名称及大小 11](#)

[5.2 查询表空间使用情况 11](#)

[5.3 查询表空间的碎片程度 11](#)

[5.4 碎片程度 12](#)

[5.5 查看回滚段名称及大小 12](#)

[5.6 查看控制文件 13](#)

[5.7 查看日志文件 13](#)

[5.8 查看表空间的使用情况 13](#)

[5.9 查看数据库对象 13](#)

[5.10 查看数据库的版本 14](#)

[5.11 查看 Oracle 字符集 14](#)

[5.12 在某个用户下找所有的索引 14](#)

[5.13 表、索引的存储情况检查 14](#)

[5.14 查看数据库的创建日期和归档方式 15](#)

[5.15 显示所有数据库对象的类别和大小 15](#)

[5.16 设置 RAC 为归档模式？ 15](#)

[6. AWR 报告 16](#)

[7. Troubleshooting 16](#)

[7.1 监控事务的等待 16](#)

[7.2 查看一些等待信息： 16](#)

[7.3 查看等待\(wait\)情况 17](#)

[7.4 回滚段查看 17](#)

[7.5 回滚段的争用情况 18](#)

| | | |
|-------|--|----|
| 7.6 | 监控表空间的 I/O 比例 | 18 |
| 7.7 | 监控文件系统的 I/O 比例 | 18 |
| 7.8 | 监控 SGA 的命中率 | 18 |
| 7.9 | 监控 SGA 中字典缓冲区的命中率 | 19 |
| 7.10 | 监控 SGA 中共享缓存区的命中率，应该小于 1% | 19 |
| 7.11 | 监控 SGA 中重做日志缓存区的命中率，应该小于 1% | 19 |
| 7.12 | 监控内存和硬盘的排序比率，最好使它小于 0.10，增加 sort area size | 20 |
| 7.13 | 监控当前数据库谁在运行什么 SQL 语句 | 20 |
| 7.14 | 监控字典缓冲区 | 20 |
| 7.15 | 查看 Lock | 20 |
| 7.16 | 捕捉运行很久的 SQL | 22 |
| 7.17 | 查看数据表的参数信息 | 22 |
| 7.18 | 查看还没提交的事务 | 23 |
| 7.19 | 查找 object 为哪些进程所用 | 23 |
| 7.20 | 查看 caught object | 23 |
| 7.21 | 查看 V\$SQLAREA | 24 |
| 7.22 | 有关 connection 的相关信息 | 24 |
| 8. | 备份 | 26 |
| 8.1 | 数据逻辑备份 | 26 |
| 8.1.1 | exp | 26 |
| 8.1.2 | imp | 27 |
| 8.2 | 控制文件备份 | 28 |
| 8.3 | 初始参数备份 | 28 |
| 8.4 | 其它 | 28 |
| 9. | 常见问题解决 | 28 |
| 9.1 | 安装后常用参数设置 | 28 |
| 9.2 | 杀死僵死连接 | 29 |
| 9.3 | AIX 内存溢出 | 30 |
| 9.4 | 某一功能特别慢 | 32 |
| 9.5 | 统计信息失效导致执行计划走全表扫描 | 32 |
| 9.6 | 中银 | 32 |
| 9.7 | 日志文件太小引起的切换过于频繁 | 35 |
| 9.8 | Oracle 连接中断问题 | 37 |
| 9.9 | 查询委托返回记录不对 | 38 |
| 9.10 | Linux + Oracle 10g RAC 的平台上，发生节点重启故障 | 39 |
| 9.11 | 联机日志损坏 | 41 |
| 9.12 | 控制文件损坏 | 41 |
| 9.13 | ORA-01555 错误 | 41 |
| 9.14 | HP Proliant DL585G2 机器安装 | 42 |

1. 前言：

有一定 Linux/Unix 操作系统、Oracle 数据库基础的工程人员和维护人员，证券公司信息技术人员等。本手册可作为工具，Oracle 运维提供帮助。

2. 简单命令使用

2.1 进入 SQL*Plus

```
$sqlplus 用户名/密码
```

2.2 退出 SQL*Plus

```
SQL>exit
```

2.3 在 sqlplus 下得到帮助信息

列出全部 SQL 命令和 SQL*Plus 命令

```
SQL>help
```

列出某个特定的命令的信息

```
SQL>help 命令名
```

2.4 显示表结构命令 DESCRIBE

```
SQL>DESC 表名
```

2.5 SQL*Plus 中的编辑命令

Ø 显示 SQL 缓冲区命令

```
SQL>L
```

Ø 修改 SQL 命令

首先要将待改正行变为当前行

```
SQL>n
```

Ø 用 CHANGE 命令修改内容

```
SQL>c/旧/新
```

Ø 重新确认是否已正确

```
SQL>L
```

Ø 使用 INPUT 命令可以在 SQL 缓冲区中增加一行或多行

```
SQL>i
```

```
SQL>输入内容
```

2.6 调用外部系统编辑器

```
SQL>edit 文件名
```

可以使用 DEFINE 命令设置系统变量 EDITOR 来改变文本编辑器的类型，在 login.sql 文件中定义如下一行

```
DEFINE _EDITOR=vi
```

2.7 运行命令文件

```
SQL>START test
```

```
SQL>@test
```

2.8 关于侦听

1、新建/修改/删除侦听

以 oracle 用户登录，运行 netca，会跳出图形配置界面。

2、打开侦听

```
lsnrctl start
```

3、查看侦听

```
Lsnrctl status
```

4、关闭侦听

```
lsnrctl stop
```

3. ORACLE 的启动和关闭

3.1 在单机环境下

要想启动或关闭 ORACLE 系统必须首先切换到 ORACLE 用户，如下

```
su - oracle
```

启动 oracle 数据库命令：

```
$sqlplus /nolog
```

```
SQL*Plus: Release 10.2.0.1.0 - Production on 星期一 7月 16 16:09:40 2007
```

```
Copyright (c) 1982, 2005, Oracle. All rights reserved.
```

```
SQL> conn / as sysdba
```

已连接到空闲例程。

```
SQL> startup
```

ORACLE 例程已经启动。

Total System Global Area 369098752 bytes

Fixed Size 1249080 bytes

Variable Size 201326792 bytes

Database Buffers 159383552 bytes

Redo Buffers 7139328 bytes

数据库装载完毕。

数据库已经打开。

关闭 oracle 数据库命令：

```
$ sqlplus /nolog
```

SQL*Plus: Release 10.2.0.1.0 - Production on 星期一 7 月 16 16:08:10 2007
Copyright (c) 1982, 2005, Oracle. All rights reserved.

```
SQL> conn / as sysdba
```

已连接。

```
SQL> shutdown immediate
```

数据库已经关闭。

已经卸载数据库。

ORACLE 例程已经关闭。

```
SQL>
```

3.2 在双机环境下

要想启动或关闭 crs 服务必须首先切换到 root 用户，如下

```
su - root
```

启动 crs 服务：

Ø 启动 CRS

```
#$CRS_HOME/crs/bin/crsctl start crs
```

Ø 查看 CRS 状态

```
#$CRS_HOME/crs/bin/crsctl check crs
```

Ø 关闭 CRS

```
#$CRS_HOME/crs/bin/crsctl stop crs
```

Ø 查看 CRS 内部各资源状态

```
#$CRS_HOME/crs/bin/crs_stat -t
```

启动数据库服务

```
# srvctl start database -d tdb#tdb 为数据库名
```

4. 数据库管理员日常工作

4.1 检查 alterSID.log

这个日志文件位于参数 BACKGROUND_DUMP_DEST 指定的目录，可能通过以下命令来查看。

```
SQL> SHOW PARAMETER background_dump_dest
```

在出现大故障前，数据库有可能会报一些警告或错误信息，应该充分重视这些信息，未雨绸缪，避免更大错误的发生。

检查 alterSID.log 的什么内容。

Ø 检查数据库是否出现过宕机（可能在晚间重启而维护人员不知道）

Ø Oracle 出错信息，通过 \$grep ORA- alterSID.log 查找

Ø 产品有关的问题：ORA-00600/ORA-07445 等错误

Ø 相应的 TRACE 文件

4.2 环境确认

数据库实例是否正常工作

```
SQL > select status from v$instance;
```

数据库监听器是否正常工作

- \$ lsnrctl status

是否存在故障表空间

- SQL> select tablespace_name,status from dba_tablespace;

控制文件、日志文件是否正常

```
SQL> select * from v$controlfile;
```

```
SQL> select * from v$log;
```

```
SQL> select * from v$logfile;
```

性能监测

Ø 每天按业务峰值情况，对数据库性能数据进行定时采集

Ø 每天检查数据库的主要性能指标

Ø 每天检查最消耗资源的 SQL 语句变化情况。

Ø 每天检查是否有足够的资源

Ø 检查所有表空间的剩余情况

Ø 识别出一些异常的增长

Ø 检查 CPU、内存、网络等是否异常

5. 数据库日常操作 SQL

5.1 查看表空间物理文件的名称及大小

```
select tablespace_name,  
       file_id,  
       file_name,  
       round(bytes / (1024 * 1024), 0) filesize  
from dba_data_files  
order by tablespace_name;
```

5.2 查询表空间使用情况

```
select a.tablespace_name "表空间名称",  
       100 - round((nvl(b.bytes_free, 0) / a.bytes_alloc) * 100, 2) "占用率  
(%)",  
       round(a.bytes_alloc / 1024 / 1024, 2) "容量(M)",  
       round(nvl(b.bytes_free, 0) / 1024 / 1024, 2) "空闲(M)",  
       round((a.bytes_alloc - nvl(b.bytes_free, 0)) / 1024 / 1024, 2) "使用  
(M)",  
       to_char(sysdate, 'yyyy-mm-dd hh24:mi:ss') "采样时间"  
from (select f.tablespace_name,  
            sum(f.bytes) bytes_alloc,  
            sum(decode(f.autoextensible, 'YES', f.maxbytes, 'NO', f.byte
```



```

s)) maxbytes
    from dba_data_files f
        group by tablespace_name) a,
    (select f.tablespace_name, sum(f.bytes) bytes_free
    from dba_free_space f
        group by tablespace_name) b
where a.tablespace_name = b.tablespace_name
order by 2 desc;

```

5.3 查询表空间的碎片程度

```

select tablespace_name, count(tablespace_name)
    from dba_free_space
    group by tablespace_name
having count(tablespace_name) > 10;

```

```

alter tablespace HS_USER_DATA coalesce;
alter table name deallocate unused;

```

5.4 碎片程度

```

select tablespace_name, count(tablespace_name)
    from dba_free_space
    group by tablespace_name
having count(tablespace_name) > 10;
alter tablespace name coalesce;
alter table name deallocate unused;
create or replace view ts_blocks_v as
    select tablespace_name, block_id, bytes, blocks, segment_name
        from dba_free_space
    union all
    select tablespace_name, block_id, bytes, blocks, segment_name
        from dba_extents;
select * from ts_blocks_v;
select tablespace_name, sum(bytes), max(bytes), count(block_id)
    from dba_free_space
    group by tablespace_name;

```

查看碎片程度高的表

```

SELECT segment_name table_name, COUNT(*) extents
    FROM dba_segments
    WHERE owner NOT IN ('SYS', 'SYSTEM')
    GROUP BY segment_name
    HAVING COUNT(*) = (SELECT MAX(COUNT(*))
        FROM dba_segments
        GROUP BY segment_name);

```

5.5 查看回滚段名称及大小

```
select segment_name,  
       tablespace_name,  
       r.status,  
       (initial_extent / 1024) InitialExtent,  
       (next_extent / 1024) NextExtent,  
       max_extents,  
       v.curext CurExtent  
  From dba_rollback_segs r, v$rollstat v  
 Where r.segment_id = v.usn(+)  
 order by segment_name;
```

5.6 查看控制文件

```
select name from v$controlfile;
```

5.7 查看日志文件

```
select member from v$logfile;
```

5.8 查看表空间的使用情况

```
select sum(bytes) / (1024 * 1024) as free_space, tablespace_name  
  from dba_free_space  
 group by tablespace_name;
```

```
SELECT A.TABLESPACE_NAME,  
       A.BYTES TOTAL,  
       B.BYTES USED,  
       C.BYTES FREE,  
       (B.BYTES * 100) / A.BYTES "% USED",  
       (C.BYTES * 100) / A.BYTES "% FREE"  
  FROM SYS.SM$TS_AVAIL A, SYS.SM$TS_USED B, SYS.SM$TS_FREE  
 C  
 WHERE A.TABLESPACE_NAME = B.TABLESPACE_NAME  
       AND A.TABLESPACE_NAME = C.TABLESPACE_NAME;
```

5.9 查看数据库对象

```
select owner, object_type, status, count(*) count#  
  from all_objects  
 group by owner, object_type, status;
```

5.10 查看数据库的版本

```
Select version
FROM Product_component_version
Where SUBSTR(PRODUCT, 1, 6) = 'Oracle';
```

5.11 查看 Oracle 字符集

```
select * from sys.props$ where name = 'NLS_CHARACTERSET';
```

5.12 在某个用户下找所有的索引

```
select user_indexes.table_name,
       user_indexes.index_name,
       uniqueness,
       column_name
from user_ind_columns, user_indexes
where user_ind_columns.index_name = user_indexes.index_name
and user_ind_columns.table_name = user_indexes.table_name
order by user_indexes.table_type,
         user_indexes.table_name,
         user_indexes.index_name,
         column_position;
```

5.13 表、索引的存储情况检查

```
select segment_name, sum(bytes), count(*) ext_quan
from dba_extents
where tablespace_name = '&tablespace_name'
and segment_type = 'TABLE'
group by tablespace_name, segment_name;
select segment_name, count(*)
from dba_extents
where segment_type = 'INDEX'
and owner = '&owner'
group by segment_name;
```

5.14 查看数据库的创建日期和归档方式

```
Select Created, Log_Mode, Log_Mode From V$Database;
```

5.15 显示所有数据库对象的类别和大小

```
select type,
       count(name) num_instances,
```

```

sum(source_size) source_size,
sum(parsed_size) parsed_size,
sum(code_size) code_size,
sum(error_size) error_size,
sum(source_size) + sum(parsed_size) + sum(code_size) +
sum(error_size) size_required
from dba_object_size
group by type
order by 1;

```

5.16 设置 RAC 为归档模式?

步骤:

1. 以 SYSDBA 身份登陆 2 个节点，执行
`alter system set cluster_database=false scope =spfile sid='*';`

设置归档路径

```
alter system set log_archive_start=true scope=spfile;
```

2. 2 个节点
`shutdown immediate`

3. 在一个节点上执行

```
startup mount
```

```
alter database archivelog;
```

```
shutdown immediate;
```

```
alter database open;
```

```
alter system set cluster_database=true scope =spfile sid='*';
```

```
shutdown immediate
```

- 4、分别启动 2 个节点，修改完毕

6. AWR 报告

与 9i 中的 statspack 相似，awr 报告也需要两个快照，才能生成这两个时间点之间的性能报告。

```
$sqlplus / as sysdba
```

Ø 生成快照一（10g 中自动会每个整点都会生成一个快照）

```
SQL> exec dbms_workload_repository.create_snapshot();
```

Ø (间隔一段时间)生成快照二

```
SQL> exec dbms_workload_repository.create_snapshot();
```

Ø 生成报告

```
SQL> @?/rdbms/admin/awrrpt.sql
```

7. Troubleshooting

常用性能相关 SQL，监控数据库性能的 SQL 语句。

7.1 监控事务的等待

```
select event,  
       sum(decode(wait_Time, 0, 0, 1)) "Prev",  
       sum(decode(wait_Time, 0, 1, 0)) "Curr",  
       count(*) "Total"  
from v$session_Wait  
group by event  
order by 4;
```

7.2 查看一些等待信息：

```
select sid, event  
from v$session_wait  
where event not like 'SQL%'  
and event not like '%ipc%';
```

查看是否存在下面等常见的等待事件：

Ø buffer busy waits,

Ø free buffer waits,

Ø db file sequential read,

Ø db file scattered read,

Ø enqueue,latch free,

Ø log file parallel write,

Ø log file sync

7.3 查看等待(wait)情况

```
SELECT v$waitstat.class,  
       v$waitstat.count count,  
       SUM(v$sysstat.value) sum_value  
FROM v$waitstat, v$sysstat  
WHERE v$sysstat.name IN ('db block gets', 'consistent gets')  
group by v$waitstat.class, v$waitstat.count;
```

7.4 回滚段查看

```
select rownum,  
       sys.dba_rollback_segs.segment_name Name,  
       v$rollstat.extents Extents,  
       v$rollstat.rssize Size_in_Bytes,  
       v$rollstat.xacts XActs,  
       v$rollstat.gets Gets,  
       v$rollstat.waits Waits,  
       v$rollstat.writes Writes,  
       sys.dba_rollback_segs.status status  
from v$rollstat, sys.dba_rollback_segs, v$rollname  
where v$rollname.name(+) = sys.dba_rollback_segs.segment_name  
      and v$rollstat.usn(+) = v$rollname.usn  
order by rownum;
```

7.5 回滚段的争用情况

```
select name, waits, gets, waits / gets "Ratio"  
from v$rollstat a, v$rollname b  
where a.usn = b.usn;■
```

7.6 监控表空间的 I/O 比例

```
select df.tablespace_name name,  
       df.file_name "file",  
       f.phyrds pyr,  
       f.phyblkrd pbr,  
       f.phywrt pyw,  
       f.phyblkwrt pbw  
from v$filestat f, dba_data_files df  
where f.file# = df.file_id  
order by df.tablespace_name;■
```

7.7 监控文件系统的 I/O 比例

```
select substr(a.file#, 1, 2) "#",  
       substr(a.name, 1, 30) "Name",  
       a.status,
```

```

a.bytes,
  b.phyrds,
  b.phywrt
from v$datafile a, v$filestat b
where a.file# = b.file#;

```

7.8 监控 SGA 的命中率

```

select a.value + b.value "logical_reads",
       c.value "phys_reads",
       round(100 * ((a.value + b.value) - c.value) / (a.value + b.value)) "BUFF
ER HIT RATIO"
from v$sysstat a, v$sysstat b, v$sysstat c
where a.statistic# = 38 -- physical read total multi block requests
and b.statistic# = 39 -- physical read total bytes
and c.statistic# = 40; -- physical write total IO requests

```

7.9 监控 SGA 中字典缓冲区的命中率

```

select parameter,
       gets,
       Getmisses,
       getmisses / (gets + getmisses) * 100 "miss ratio",
       (1 - (sum(getmisses) / (sum(gets) + sum(getmisses)))) * 100 "Hit ratio"
from v$rowcache
where gets + getmisses <> 0
group by parameter, gets, getmisses;

```

7.10 监控 SGA 中共享缓存区的命中率，应该小于 1%

```

select sum(pins) "Total Pins",
       sum(reloads) "Total Reloads",
       sum(reloads) / sum(pins) libcache
from v$librarycache;
select sum(pinhits - reloads) / sum(pins) * 100 "hit radio",
       sum(reloads) / sum(pins) "reload percent"
from v$librarycache;

```

7.11 临控 SGA 中重做日志缓存区的命中率，应该小于 1%

```

SELECT name,
       gets,
       misses,
       immediate_gets,

```

```

        immediate_misses,
        Decode(gets, 0, 0, misses / gets * 100) ratio1,
        Decode(immediate_gets + immediate_misses,
        0,
        0,
        immediate_misses / (immediate_gets + immediate_misses) *
100) ratio2
    FROM v$latch
    WHERE name IN ('redo allocation', 'redo copy');

```

7.12 监控内存和硬盘的排序比率，最好使它小于 0.10，增加 sort_area_size

```

SELECT name, value
  FROM v$sysstat
 WHERE name IN ('sorts (memory)', 'sorts (disk)');

```

7.13 监控当前数据库谁在运行什么 SQL 语句

```

SELECT osuser, username, sql_text
  from v$session a, v$sqltext b
 where a.sql_address = b.address
 order by address, piece;

```

7.14 监控字典缓冲区

```

SELECT SUM(PINS) "EXECUTIONS",
       SUM(RELOADS) "CACHE MISSES WHILE EXECUTING",
       (SUM(PINS - RELOADS)) / SUM(PINS) "LIB CACHE"
  FROM V$LIBRARYCACHE;

SELECT SUM(GETS) "DICTIONARY GETS",
       SUM(GETMISSES) "DICTIONARY CACHE GET MISSES",
       (SUM(GETS - GETMISSES - USAGE - FIXED)) / SUM(GETS) "ROW
CACHE"
  FROM V$ROWCACHE;

```

“LIB CACHE”与“ROW CACHE”越接近 1.00 超好，不要低于 0.90。否则需要调大 SGA 的空间。

7.15 查看 Lock

```

select s.osuser,
       l.sid,
       s.serial#,
       s.username,

```



```

s.terminal,
  decode(l.type,
    'TM',
    'TM - DML Enqueue',
    'TX',
    'TX - Trans Enqueue',
    'UL',
    'UL - User',
    l.type || ' - Other Type') LOCKTYPE,
substr(t.name, 1, 10) OBJECT,
u.name owner,
l.id1,
l.id2,
decode(l.lmode,
  1,
  'No Lock',
  2,
  'Row Share',
  3,
  'Row Exclusive',
  4,
  'Share',
  5,
  'Shr Row Excl',
  6,
  'Exclusive',
  null) lmode,
decode(l.request,
  1,
  'No Lock',
  2,
  'Row Share',
  3,
  'Row Excl',
  4,
  'Share',
  5,
  'Shr Row Excl',
  6,
  'Exclusive',
  null) request
from v$lock l, v$session s, sys.user$ u, sys.obj$ t
where l.sid = s.sid
and s.type != 'BACKGROUND'

```

```
and t.obj# = l.id1
and u.user# = t.owner#;
```

7.16 捕捉运行很久的 SQL

```
select username,
       sid,
       opname,
       round(sofar * 100 / totalwork, 0) || '%' as progress,
       time_remaining,
       sql_text
from v$session_longops, v$sql
where time_remaining <> 0
and sql_address = address
and sql_hash_value = hash_value;
```

7.17 查看数据表的参数信息

```
SELECT partition_name,
       table_name,
       high_value,
       high_value_length,
       tablespace_name,
       pct_free,
       pct_used,
       ini_trans,
       max_trans,
       initial_extent,
       next_extent,
       min_extent,
       max_extent,
       pct_increase,
       FREELISTS,
       freelist_groups,
       LOGGING,
       BUFFER_POOL,
       num_rows,
       blocks,
       empty_blocks,
       avg_space,
       chain_cnt,
       avg_row_len,
       sample_size,
       last_analyzed
FROM sys.dba_tab_partitions
WHERE table_owner = 'HS_HIS' AND table_name = 'HISBANKTRADE'
```

```
--WHERE table_name = :tname AND table_owner = :towner  
ORDER BY partition_position,partition_name;
```

7.18 查看还没提交的事务

```
select * from v$locked_object;  
select * from v$transaction;
```

7.19 查找 object 为哪些进程所用

```
select p.spid,  
       s.sid,  
       s.serial# serial_num,  
       s.username user_name,  
       a.type object_type,  
       s.osuser os_user_name,  
       a.owner,  
       a.object object_name,  
       decode(sign(48 - command),  
              1,  
              to_char(command),  
              'Action Code #') || to_char(command)) action,  
       p.program oracle_process,  
       s.terminal terminal,  
       s.program program,  
       s.status session_status  
from v$session s, v$access a, v$process p  
where s.paddr = p.paddr  
      and s.type = 'USER'  
      and a.sid = s.sid  
      and a.object = 'FUNDREAL'  
order by s.username, s.osuser;
```

7.20 查看 caught object

```
SELECT owner,  
       name,  
       db_link,  
       namespace,  
       type,  
       sharable_mem,  
       loads,  
       executions,  
       locks,  
       pins,  
       kept  
FROM v$db_object_cache where owner LIKE 'HS_%';
```

7.21 查看 V\$SQLAREA

```
SELECT SQL_TEXT,  
       SHARABLE_MEM,  
       PERSISTENT_MEM,  
       RUNTIME_MEM,  
       SORTS,  
       VERSION_COUNT,  
       LOADED_VERSIONS,  
       OPEN_VERSIONS,  
       USERS_OPENING,  
       EXECUTIONS,  
       USERS_EXECUTING,  
       LOADS,  
       FIRST_LOAD_TIME,  
       INVALIDATIONS,  
       PARSE_CALLS,  
       DISK_READS,  
       BUFFER_GETS,  
       ROWS_PROCESSED  
FROM V$SQLAREA;
```

7.22 有关 connection 的相关信息

I 查看有哪些用户连接

```
select s.sid,  
       s.serial# serial_num,  
       s.osuser os_user_name,  
       decode(sign(48 - command),  
              1,  
              to_char(command),  
              'Action Code #' || to_char(command)) action,  
       p.program oracle_process,  
       status session_status,  
       s.terminal terminal,  
       s.program program,  
       s.username user_name,  
       s.fixed_table_sequence activity_meter  
from v$session s, v$process p  
where s.paddr = p.paddr  
and s.type = 'USER' order by s.username, s.osuser;
```

l 2)根据 v.sid 查看对应连接的资源占用等情况

```
select n.name, v.value, n.class, n.statistic#  
  from v$statname n, v$sesstat v  
 where v.sid = &sid  
    and v.statistic# = n.statistic#  
 order by n.class, n.statistic#;
```

l 3)根据 sid 查看对应连接正在运行的 sql

```
select /*+ PUSH_SUBQ */  
command_type,  
sql_text,  
sharable_mem,  
persistent_mem,  
runtime_mem,  
sorts,  
version_count,  
loaded_versions,  
open_versions,  
users_opening,  
executions,  
users_executing,  
loads,  
first_load_time,  
invalidations,  
parse_calls,  
disk_reads,  
buffer_gets,  
rows_processed,  
sysdate start_time,  
sysdate finish_time,  
> || address sql_address,  
'N' status  
  from v$sqlarea  
 where address = (select sql_address from v$session where sid = &sid);
```

8. 备份

8.1 数据逻辑备份

8.1.1 exp

将数据库内的各对象以二进制方式下载成 dmp 文件，方便数据迁移。

buffer: 下载数据缓冲区，以字节为单位，缺省依赖操作系统
consistent: 下载期间所涉及的数据保持 read only，缺省为 n
direct: 使用直通方式，缺省为 n
feedback: 显示处理记录条数，缺省为 0，即不显示
file: 输出文件，缺省为 expdat.dmp
filesize: 输出文件大小，缺省为操作系统最大值
indexes: 是否下载索引，缺省为 n，这是指索引的定义而非数据
log: log 文件，缺省为无，在标准输出显示
owner: 指明下载的用户名
query: 选择记录的一个子集
rows: 是否下载表记录
tables: 输出的表名列表
Ø 下载整个实例

```
exp hs_user/handsome file=oradb.dmp log=oradb.log full=y consistent=y direct=y
```

user 应具有 dba 权限

Ø 下载某个用户所有对象

```
exp hs_user/handsome file=dbuser.dmp log=dbuser.log owner=dbuser buffer=4096000 feedback=10000
```

Ø 下载一张或几张表

```
exp hs_user/handsome file=dbuser.dmp log=dbuser.log tables=table1,table2 buffer=4096000 feedback=10000
```

Ø 下载某张表的部分数据(query 选项)

```
exp hs_user/handsome file=dbuser.dmp log=dbuser.log tables=table1 buffer=4096000 feedback=10000 query=\"where col1='...' and col2 '<...\"
```

不可用于嵌套表

Ø 以多个固定大小文件方式下载某张表

```
exp hs_user/handsome file=1.dmp,2.dmp,3.dmp,... filesize=1000m tables=emp buffer=4096000 feedback=10000
```

这种做法通常用在：表数据量较大，单个 dump 文件可能会超出文件系统的限制

Ø 直通路径方式

```
direct=y，则会取代 buffer 选项。同时，query 选项不可用。
```

有利于提高下载速度

Ø consistent 选项

自 export 启动后，consistent=y 冻结来自其它会话的对 export 操作的数据对象的更新，这样可以保证 dump 结果的一致性。但这个过程不能太长，以免回滚段和联机日志消耗完

8.1.2 imp

将 exp 下载的 dmp 文件上载到数据库内。

buffer: 上载数据缓冲区，以字节为单位，缺省依赖操作系统

commit: 上载数据缓冲区中的记录上载后是否执行提交

feedback: 显示处理记录条数，缺省为 0，即不显示

file: 输入文件，缺省为 expdat.dmp

filesize: 输入文件大小，缺省为操作系统最大值

fromuser: 指明来源用户方

ignore: 是否忽略对象创建错误，缺省为 n

indexes: 是否上载索引，缺省为 n，这是指索引的定义而非数据，如果上载时索引已建立，此选项即使为 n 也无效，imp 自动更新索引数据

log: log 文件，缺省为无，在标准输出显示

rows: 是否上载表记录

tables: 输入的表名列表

touser: 指明目的用户方

Ø 上载整个实例

```
imp hs_user/handsome file=oradb.dmp log=oradb.log full=y buffer=4096000 c
ommit=y ignore=y feedback=10000
```

Ø 上载某个用户所有对象

```
imp hs_user/handsome file=dbuser.dmp log=dbuser.log fromuser=dbuser tou
ser=dbuser2 buffer=2048000 commit=y ignore=y feedback=10000
```

Ø 上载一张或几张表

```
imp dbuser2/oracle file=user.dmp log=user.log tables=table1,table2 fromuser
=dbuser touser=dbuser2 buffer=2048000 commit=y ignore=y feedback=1000
0
```

Ø 以多个固定大小文件方式上载某张表

```
imp hs_user/handsome file=\\(1.dmp,2.dmp,3.dmp,...\\) filesize=1000m tables=
emp fromuser=dbuser touser=dbuser2 buffer=4096000 commit=y ignore=y fe
edback=10000
```

|

8.2 控制文件备份

```
Alter system backup controlfile to trace;
```

8.3 初始参数备份

```
Create pfile='/tmp/pfile20070723.ora' from spfile;
```

8.4 其它

至于 tnsnames.ora, listener.ora 使用操作系统拷贝。

9. 常见问题解决

9.1 安装后常用参数设置

| 参数 | 说明 |
|-------------------------------------|--|
| <code>db_writer_processes</code> | 配置为 CPU 内核数的一半 |
| <code>open_cursors</code> | 根据 AS 数量及 AS 连接数，配置为 3000 ~ 5000 |
| <code>optimizer_mode</code> | RULE |
| <code>pga_aggregate_target</code> | 同样，也与根据 AS 数量及 AS 连接数相关，一般为 1500 ~ 3000。一般 1G ~ 3G |
| <code>processes</code> | 进程数，即可以对外提供服务的数据。同样，也与根据 AS 数量及 AS 连接数相关 1500 ~ 3000 |
| <code>session_cached_cursors</code> | |
| <code>sessions</code> | 设置 <code>processes</code> 后自动调整。 |
| <code>sga_max_size</code> | 内存的一半，如果存储使用的是裸设备，可以提高到 60% |
| <code>sga_target</code> | 同上；若设置为小于等于 <code>sga_max_size</code> ，可动态修改；否则则需要先设置 <code>sga_max_s</code> |
| <code>shared_pool_size</code> | 不超过 800M，一般 300M 到 500M 就足够了 |

注意：基于 Linux 平安的 RAC 不可设 sga 太大，否则可能会有问题。具体参见第 9 章 11 节。

```
SQL> sqlplus / as sysdba
```

当前参数值查看方法如下(`open_cursors`):

```
SQL> show parameter open_cursors;
```

参数设置方法如下：

```
SQL> alter system set open_cursors=4000; --非初始参数
```

```
SQL> alter system set db_writer_processes=2 scope=spfile; --初始参数，设置后需要重启数据库
```


9.2 杀死僵死连接

有时候由于客户端的异常退出，会出现一些连接的进程一直在运行，从而不能释放资源，影响系统的性能。可用以下方法实现把僵死的连接清除。

Ø 查询数据库死锁

```
select distinct *
  from (select t2.username, t2.sid, t2.serial#, t2.logon_time
        from v$locked_object t1, v$session t2
        where t1.session_id = t2.sid)
 order by logon_time;
```

Ø 查询出来的结果就是有死锁的 session 了，

```
alter system kill session 'sid,serial#';
```

Ø 一般情况可以解决数据库存在的死锁了，或通过 session id 查到对应的操作系统进程，在 unix 中杀掉操作系统的进程。

```
SELECT a.username, c.spid AS os_process_id, c.pid AS oracle_process_id
  FROM v$session a, v$process c
 WHERE c.addr = a.paddr
       and a.sid = m
       and a.serial# = n;
```

Ø 采用 kill (unix/linux) 或 orakill (windows)

```
kill -9 os_process_id
```

9.3 AIX 内存溢出

Ø 环境:

AIX 平台，任何版本的 Oracle

Ø 现象:

第一次：小机内核的 CPU 利用率达到 40%，有一个进程占用内存不释放，

Paging Space 达到 100%。重启 AS、迪斯杰备份及小机后利用率及

Paging Space 下降至正常。

第二次：闭市后用 SQLPlus 连接到 RAC1 上，做了一些三方存管帐户的批量修改。晚上在清算处理时，在文件标志复位时发现很慢，查看 Paging Space 利用率为 40%，但小机利用率正常。重开 AS 及迪斯杰备份后，利用率下降到 30%，文件标志复位也很快了，在清算过程中利用率也一直在下降，一直到 19% 点多。清算结束后又重开 AS 及迪斯杰备份，Paging Space 利用率为 18.9%，后将 RAC1 和 rac2 的 ORACLE 数据库重开（shut down 及 start）后，下降到 5.6%。

Ø 原因：

当 Sga, Pga 设置过小时，数据库会把不能置于内存中的数据放到交换区去，磁盘的 IO 速度比内存慢很多，所以会对性能产生较大影响。在极端情况下，如果交换分区全部耗尽，那对系统性能的影响就更大了。

在 AIX 系统中，内存可以分为两类：**计算型内存和文件型内存**。类似大量文件类操作，如文件复制、压缩、数据库的 dump/load 等操作会大量占用文件型内存。如果按照系统缺省的配置，文件型内存最多会占用到内存总量的 80%，由于文件型内存占用并不主动释放，从而可能造成内存资源的短缺及 Paging Space 使用率过高。特别是，由于数据库存储使用的是裸设备，允许文件缓存占用大量的内存是没有必要的。

命令 topas 可以用来查看文件型内存占用内存情况，在 MEMORY 一栏中的 %Noncomp 显示的是文件型内存的占用百分比。

如果文件型内存所占比例很高，而 Paging Space 使用率居高不下时，可以通过降低 minperm、maxperm 的参数值来进行调优，减少文件型内存可占用的份额。

Ø 解决方法：

在进行调整前，首先通过命令 `vmo -a` 来查看目前的参数值；`minperm` 及 `maxperm` 的缺省值分别是 20%和 80%。注意：在调小 `maxperm%`前，请产生同时降低 `maxclient%`的值（即 `maxclient%`的值不得大于 `maxperm%`的参数值）。谨慎起见，可以通过小量多次调整来达到性能优化的目的。如 `maxperm%` 可以依次调整了 60，40，20，同时查看性能的性能。最终可能可以调整为：

```
vmo -o minperm%=10

vmo -o maxclient%=20

vmo -o maxperm%=20
```

因为参数 `maxperm` 充当的是软限制，所以文件型内存的占用率依旧可以超过 `maxperm` 的设定值。如果需要进行强制限制，则需要将参数 `strict_maxperm` 的值设为 1，该参数的缺省值为 0，但此方法需谨慎使用。

Ø 其它

交换分区设置过小也可能引起交换分区利用率过高的问题。交换分区的大小与物理内存的大小有关。一般情况下，交换分区可以参照下表设置。

| 物理内存 (RAM) | 交换分区 (Swap Space) |
|-----------------------------|-------------------|
| Between 1024 MB and 2048 MB | 2 倍 RAM |
| Between 2049 MB and 8192 MB | 1.5 倍 RAM |
| More than 8192 MB | 1 倍 RAM |

9.4 某一功能特别慢

取故障生产间的两个快照，然后做一个 `AWR` 报靠。同时，也可以用第 7 部分 `Troubleshooting` 提供的语句对数据库进行分析。对抓出来的语句进行分析，如果怀疑是统计信息不正确引起的全表扫描，可以先进行统计信息的收集，方法如下。否则则可能需要测试索引。具体联系开发人员。

9.5 统计信息失效导致执行计划走全表扫描

Ø 基于 Schema 的统计信息收集

```
begin
  dbms_stats.gather_schema_stats(ownname      => 'HS_FUTURES',
                                  estimate_percent => 20,
                                  method_opt     => 'for all indexed columns size 10',
                                  cascade        => true,
                                  degree         => 7);
end;
```

Ø 基于表的统计信息收集

Analyze table hs_secu.entrust compute statistics;

9.6 关于 DRM

Ø 环境:

10g, 至少 HP-UN

Ø 现象:

在 2007-05-21 14:42:00 左右, 管理员发现报盘系统 hang 住, AS 队列上有大量积压, 但未登陆到 oracle 系统查看等待事件, 也未作其他的数据库方面的诊断处理。sqlplus 正常登陆但比较缓慢。在操作系统层面, 节点 1 的 cpu 利用率为 0, 节点 2 的 cpu 利用率为 20。管理员在反复关闭并重开报盘系统 3 分钟左右后, 系统正常运行。

事后, HP 工程师在主机层面查看了相关的日志信息, 并未发现什么异常的情况, 同时表示无法在现有资源下监控数据库数据线网络的流量。查看了 oracle 的相关日志, 发现里面有大量的关于 DRM 的信息, 根据里面的信息, 我猜测可能是由于数据库的对象资源动态管理过于频繁, 在两个实例之间来回“震荡”时出现异常的缘故。

Ø 原因:

由于在 oracle 的日志和跟踪文件里未能找到与 DRM 相关的以外的其他有用信息。根据 Oracle 工程师的反馈, 可以对 DRM 做一些处理。并表示对系统性能并无什么影响。再者, 从 Oracle 的 trace 文件里可以看出在两次交易故障发生时, oracle rac 的 DRM 正在做大量的数据同步信息, 可查看

pdcrac1_lmon_18807.trc。因此, 综合以上的各种信息, 有必要现在针对 DRM 方面做一些处理, 修改以下初始化参数。

`_gc_affinity_time=0` # defines the frequency in minutes to check if remastering is needed.

`_gc_undo_affinity=FALSE` # defines the number of times a node must access an object for it to be a DRM candidate

Ø 解决方法:

1. 备份当前的数据库服务器参数

```
SQL> Create pfile='home/oracle/initsid.ora' from spfile;
```

2. 修改备份出来的数据库参数文件, 在文件末尾添加

```
*._gc_affinity_time=0
```

```
*._gc_undo_affinity=false
```

3. shutdown 当前数据库, 已刚才备份修改后的数据库参数文件启动到 nomount

```
SQL> Shutdown immediate (两台机)
```

```
SQL> Startup pfile='/home/oracle/initsid.ora' nomount;
```

```
SQL> Create spfile='/dev/raw/rspfile' from pfile='/home/oracle/initsid.ora';
```

4. shutdown 当前数据库并重起数据库

```
SQL> Startup
```

Ø 其它

下面是 trace 文件的部分内容。

kjfcfrfg: DRM window size = 0->4096 (min lognb = 15)

*** 2007-05-19 15:28:08.141

Reconfiguration started (old inc 0, new inc 16)

Synchronization timeout interval: 900 sec

List of nodes:

0 1

*** 2007-05-19 15:28:08.141

Global Resource Directory frozen

node 0

release 10 2 0 2

node 1

release 10 2 0 2

number of mastership buckets = 128

domain attach called for domid 0

* kjbdomalc: domain 0 invalid = TRUE

* kjbdomatt: first attach for domain 0

asby init, 0/0/x1

asby returns, 0/0/x1/false

* Domain maps before reconfiguration:

* DOMAIN 0 (valid 0): 0

* End of domain mappings

* Domain maps after recomputation:

* DOMAIN 0 (valid 0): 0 1

* End of domain mappings

Dead inst

Join inst 0 1

Exist inst

Active Sendback Threshold = 50 %

Communication channels reestablished

sent syncr inc 16 lvl 1 to 0 (16,5/0/0)

sent synca inc 16 lvl 1 (16,5/0/0)

received all domreplay (16.6)

sent master 1 (16.6)

sent syncr inc 16 lvl 2 to 0 (16,7/0/0)

sent synca inc 16 lvl 2 (16,7/0/0)

Master broadcasted resource hash value bitmaps

* kjfcrfg: domain 0 valid, valid_ver = 16

Non-local Process blocks cleaned out

Set master node info

sent syncr inc 16 lvl 3 to 0 (16,13/0/0)

sent synca inc 16 lvl 3 (16,13/0/0)

Submitted all remote-enqueue requests

kjfcrfg: Number of mesgs sent to node 1 = 0

sent syncr inc 16 lvl 4 to 0 (16,15/0/0)

sent synca inc 16 lvl 4 (16,15/0/0)

Dwn-cvts replayed, VALBLKs dubious

sent syncr inc 16 lvl 5 to 0 (16,18/0/0)

sent synca inc 16 lvl 5 (16,18/0/0)

All grantable enqueues granted

sent syncr inc 16 lvl 6 to 0 (16,20/0/0)

sent synca inc 16 lvl 6 (16,20/0/0)

*** 2007-05-19 15:28:09.638

Submitted all GCS cache requests

sent syncr inc 16 lvl 7 to 0 (16,22/0/0)

sent synca inc 16 lvl 7 (16,22/0/0)

Post SMON to start 1st pass IR

Fix write in gcs resources

sent syncr inc 16 lvl 8 to 0 (16,24/0/0)

sent synca inc 16 lvl 8 (16,24/0/0)

*** 2007-05-19 15:28:12.962

Reconfiguration complete

* domain 0 valid?: 1

*** 2007-05-19 15:28:18.377

kjxgrtmc2: mounting member 0 thread 1

Begin DRM(32)

sent syncr inc 16 lvl 9 to 0 (16,0/31/0)

sent synca inc 16 lvl 9 (16,0/31/0)

...

sent syncr inc 16 lvl 39 to 0 (16,0/36/0)

sent synca inc 16 lvl 39 (16,0/36/0)

*** 2007-05-19 15:28:28.459

sent syncr inc 16 lvl 40 to 0 (16,0/38/0)

sent synca inc 16 lvl 40 (16,0/38/0)

End DRM(32)

9.7 日志文件太小引起的切换过于频繁

假设现有三个日志组，每个组内有一个成员，每个成员的大小为 512MB，现在想把此三个日志组的成员大小都改为 1G。

Ø 创建 2 个新的日志组

```
alter database add logfile group 4 ('/oracle/oradata/redo04_1.log') size 1024M
;
alter database add logfile group 5 ('/oracle/oradata/redo05_1.log') size 1024M
;
```

Ø 切换当前日志到新的日志组

```
alter system switch logfile;
```

```
alter system switch logfile;
```

Ø 删除旧的日志组

```
alter database drop logfile group 1;
```

```
alter database drop logfile group 2;
```

```
alter database drop logfile group 3;
```

Ø 操作系统下删除原日志组 1、2、3 中的文件

Ø 重建日志组 1、2、3

```
alter database add logfile group 1 ('/oracle/oradata/redo01_1.log') size 1024M
;
```

```
alter database add logfile group 2 ('/oracle/oradata/redo02_1.log') size 1024M
;
```

```
alter database add logfile group 3 ('/oracle/oradata/redo03_1.log') size 1024M
;
```

Ø 切换日志组

```
alter system switch logfile;
```

```
alter system switch logfile;
```

```
alter system switch logfile;
```

Ø 删除中间过渡用的日志组 4、5

```
alter database drop logfile group 4;
```

```
alter database drop logfile group 5;
```

Ø 到操作系统下删除原日志组 4、5 中的文件

Ø 备份当前的最新的控制文件

```
SQL> connect internal
```

```
SQL> alter database backup controlfile to trace resetlogs
```

9.8 Oracle 连接中断问题

Ø 环境:

10201, AIX53 (据 ORACLE 称, 在任何操作系统版本都有此问题)

Ø 现象:

监听器启动后, 隔一段时间 (长短不定), 就会出现无法连接:

若是用 10201 版本的 SQLPLUS, 则会出现 NO LISTENER。

若是用 9207 版本的 SQLPLUS, 则会出现: 没反应, HANG 住。

Ø 原因:

10201 版本上的一个 BUG：4518443。其会自动创建一个子监听器，当出现此情况时，监听器将会挂起。

检查是否真因为此 BUG 造成此现象：

```
$ ps -ef | grep tnslnr
```

```
ora10g 8909 1 0 Sep 15 ? 902:44 /u05/10GHOME/DBHOME/bin/tnslnr sales  
-inherit
```

```
ora10g 22685 8909 0 14:19:23 ? 0:00 /u05/10GHOME/DBHOME/bin/tnslnr s  
ales -inherit
```

正常情况只有一个监听器，而此 BUG 则会出现两个监听器。

Ø 解决方法：

打补丁 4518443 或者在 listener.ora 文件里加入：

```
SUBSCRIBE_FOR_NODE_DOWN_EVENT_<listener_name>=OFF
```

其中，<listener_name> 是数据库的监听器的名称。如：

默认情况下，监听器名为：LISTENER。则语句就是：

```
SUBSCRIBE_FOR_NODE_DOWN_EVENT_LISTENER=OFF
```

9.9 查询委托返回记录不对

Ø 环境：

10g，任何平台

Ø 现象：

查询语句结果集返回不全，只返回一条或其中几条。

Ø 原因:

未知, 怀疑为 Oracle Bug

Ø 解决方法:

收集相关表的统计信息, 如:

```
SQL> Analyze table hs_secu.entrust compute statistics;
```

如问题仍然存在, 使用下面语句:

```
SQL> Analyze table hs_secu.entrust compute statistics for all indexes;
```

Ø 其它

这个问题已经多次出现, 与很多 Oracle 售前人员交流过, 没有定论。目前怀疑与下面这个 Bug 相关。具体内容参见 metalink:

Wrong Results Possible on 10.2 When New "HASH GROUP BY" Feature is Used ([Doc ID](#): Note:387958.1)

I Cause

[Bug 4604970](#) Wrong Results With 'Hash Group By' Aggregation Enabled
Any query that displays "HASH GROUP BY" in the execution plan could hit this bug.

The "HASH GROUP BY" feature is enabled by default in 10.2.

I Solution

Patch 4604970 can be downloaded from Metalink for most platforms .

Problem is resolved in latest Patchset release 10.2.0.3.

Workaround:

Increase the memory used by hash group by

Or

Disable the use of hash group by

eg: set optimizer_features_enable to "10.1.0"

or

set "_gby_hash_aggregation_enabled" to FALSE.

9.10 Linux + Oracle 10g RAC 的平台上，发生节点重启故障

Ø 环境:

I 服务器硬件平台:

- HP DL580 G4 4CPU (双核) 16G 内存

- IBM 3950 4CPU (双核) 16G 内存

I 操作系统平台:

- Redhat Linux AS 4 Update 4 64bit

I 数据库系统:

- Oracle 10g R2(10.2.0.1) RAC (有 2 节点的, 4 节点的)

Ø 现象:

其中某一结点操作系统重启。经典场景: 前一日清算到资金清算汇总时 rac 节点 2 自动重启, 次日 9:10 rac 节点 1 死机, 冷启动后恢复正常。

I ALERT.LOG 日志

Mon Jun 11 16:54:38 2007

Error: unexpected error (6) from the Cluster Service (LCK0)

Mon Jun 11 16:54:38 2007

Errors in file /u01/app/oracle/admin/rac/bdump/rac1_lck0_9207.trc:

ORA-29702: error occurred in Cluster Group Service operation

Mon Jun 11 16:54:38 2007

Errors in file /u01/app/oracle/admin/rac/bdump/rac1_lms3_9100.trc:

ORA-29702: error occurred in Cluster Group Service operation

Mon Jun 11 16:54:38 2007

Errors in file /u01/app/oracle/admin/rac/bdump/rac1_lms0_9088.trc:

ORA-29702: error occurred in Cluster Group Service operation

Mon Jun 11 16:54:38 2007

Errors in file /u01/app/oracle/admin/rac/bdump/rac1_lmd0_9086.trc:

ORA-29702: error occurred in Cluster Group Service operation

Mon Jun 11 16:54:38 2007

Errors in file /u01/app/oracle/admin/rac/bdump/rac1_psp0_9082.trc:

ORA-29702: error occurred in Cluster Group Service operation

Mon Jun 11 16:54:38 2007

System state dump is made for local instance

Mon Jun 11 16:54:38 2007

Errors in file /u01/app/oracle/admin/rac/bdump/rac1_j000_12310.trc:

ORA-29702: error occurred in Cluster Group Service operation

Mon Jun 11 16:54:38 2007

Errors in file /u01/app/oracle/admin/rac/bdump/rac1_pmon_9078.trc:

ORA-29702: error occurred in Cluster Group Service operation

Mon Jun 11 16:54:38 2007

Errors in file /u01/app/oracle/admin/rac/bdump/rac1_lms2_9096.trc:

ORA-29702: error occurred in Cluster Group Service operation

Mon Jun 11 16:54:38 2007

Errors in file /u01/app/oracle/admin/rac/bdump/rac1_lms1_9092.trc:

ORA-29702: error occurred in Cluster Group Service operation

Mon Jun 11 16:54:38 2007

Errors in file /u01/app/oracle/admin/rac/bdump/rac1_arc1_9345.trc:

ORA-29702: error occurred in Cluster Group Service operation

Mon Jun 11 16:54:38 2007

Errors in file /u01/app/oracle/admin/rac/bdump/rac1_arc1_9345.trc:

ORA-29702: error occurred in Cluster Group Service operation

System State dumped to trace file /u01/app/oracle/admin/rac/bdump/rac1_diag_9080.trc

Mon Jun 11 16:54:38 2007

LCK0: terminating instance due to error 29702

I OCSSD.LOG

[CSSD]2007-06-

11 16:54:35.377 [1147169120] >ERROR: clssnmDiskPMT: 3 of 3 voting disks unavailable (0/0/3)

[CSSD]2007-06-

11 16:54:35.380 [1157658976] >TRACE: clssnmHandleSync: Acknowledging sync: src[2] srcName[tdb12] seq[13] sync[29]

[CSSD]2007-06-

11 16:54:35.380 [1157658976] >TRACE: clscsendx: (0x2a97c1df40) Physical connection (0x2a97c35080) not active

[CSSD]2007-06-

11 16:54:35.381 [1157658976] >WARNING: clssnmSendMsg: send failed, node 2, type 11, rc 10

Ø 原因:

未知, 怀疑为 Oracle Bug, 但是把 SGA 调小可使此问题消失。

Ø 解决方法:

问题原因还在追踪, METALINK 上并没有找到针对性的 BUG 或解决方案, 怀疑是 linux 平台下 Oracle 对 SGA 管理的问题, 很可能是个 Bug, 应该说 SGA

调小只是针对故障的临时解决方案。最终原因，将会在下一阶段测试新的版本 10.2.0.3 来确认。目前如果发生此问题，需要把 SGA 调到 6G 以下，下面为修改方法：

I 调整 linux 内核参数，编辑 /etc/sysctl.conf 文件修改以下内容

kernel.threads-max = 131072 fs.file-max = 131072

编辑上面内容保存退出后，执行 `sysctl -p` 使参数生效。

I 调整 sga 大小

SQL> Alter system set sga_max_size=6000M scope=spfile;

SQL> Alter system set sga_target=6000M scope=spfile;

然后按正常方式重启数据库。

9.11 联机日志损坏

Ø 非当前日志损坏(已归档)

Alter database clear logfile group n 来创建一个新的日志文件

Ø 非当前日志损坏(未归档)

Alter database clear unarchived logfile group n

Ø 如果是当前日志损坏，一般不能 clear，否则可能意味着丢失数据

如果有备份，可以采用备份进行不完全恢复

如果没有备份，可能只能用 `_allow_resetlogs_corruption=true` 来进行强制恢复了，但是，这样的方法是不建议的，容易导致数据字典不一致，一般这样打开以后要进行全库备份，再重新建库导出数据

9.12 控制文件损坏

Ø 1、损坏单个控制文件

关闭数据库，拷贝一个好的文件覆盖掉即可；或者是修改 init.ora 文件的相关部分

Ø 2、损失全部控制文件，

创建控制文件或从备份恢复创建控制文件的脚本可以通过
`alter database backup controlfile to trace` 获取。

9.13 ORA-01555 错误

Ø 环境:

任何平台

Ø 现象:

报 ORA-01555 错。

Ø 发生地点:

Ø 原因:

读一致性引起，一般是由于以下原因造成：

1、UNDO 太小

2、9i 里的 UNDO_RETENTION 参数设置太小，导致已经提交事务在 UNDO 里
快速被覆盖

3、查询语句效率太低

Ø 解决方法:

1、适当加大 UNDO 表空间

2、加大 UNDO_RETENTION 参数

3、调整应用程序，提交查询的效率

Ø 其它

9.14 HP Proliant DL585G2 机器安装

HP Proliant DL585G2 机器安装 RHEL AS4 时可能出现的操作系统检测不到网卡，还有插在 PCI1 和 PCI2 槽上的两块 FCA2214 HBA 卡也没有被检测出的情况的处理

Ø 解决方法：

Linux 安装完成后，在 `/etc/grub.conf` 文件中，修改启动系统的引导项，增

加参数 `"pci=nommconf"`，如：

`kernel /vmlinuz-2.6.9-`

`55.0.0.0.2.ELsmp ro root=LABEL=/ rhgb "pci=nommconf" quiet`

用网络图形配置工具“New”一个新的设备