

ParnassusData is a software company

ORACLE DB 数据库常见问题解决 及诊断技巧集锦-ORACLE DBA 故障 修复必备手册

Parnassus
诗 檀

Creation Date: June 09,2016
Last Update: June 01,2016

文档控制

变更记录

日期	作者及更新人	版本号	变更信息

审阅人

版本号	审阅人	职位	相关评论

审批人

版本号	批准人	日期	相关评论

文档分发

分发号	文档名	分发位置

目录

文档控制	2
变更记录	2
审阅人	2
审批人	2
文档分发	2
目录	3
关于 ORACLE 数据库问题诊断信息获取	5
诊断数据库挂起(HANG)事件	8
PROCESS SPINS (进程消耗 CPU 资源达 100%)	11
ORA-4031 错误分析	12
ORA-4030 报错	14
ORA-600 错误	15
ORA-1801 错误	16
ORA-7445 错误	17
其它 ORA-NNNN 报错	18
DATABASE PERFORMANCE 数据库性能问题	19
监听器挂起及性能问题	21
SQL 性能问题	24
安装/升级/重连问题	26
RAC 性能问题	27
RAC 实例驱逐问题	28
请具体描述问题. 当提到问题发生时, 请尽量精确, 如某个特定时间等.	28
ORACLE 集群就绪服务(CRS)问题	29
DATA GUARD 日志传输和物理应用	31
DATA GUARD 网络故障	32
STREAMS 错误及问题	33
STREAMS 应用(APPLY)	34
高级队列传播(AQ PROPAGATION)问题	36
复制问题(REPLICATION)	37
ASM 问题诊断 (包括在空闲空间存在并可用时的 ASM 磁盘空间耗尽报错)	38
消息网关(MESSAGE GATEWAY)问题	39
ORA-3113 错误	42
数据库/实例崩溃	43
数据库块讹误(CORRUPTION)	44
RMAN 报错	45
ORACLE 企业管理器 / 网格控制(OEM/GRID CONTROL)	46

SQL*NET 报错.....	47
补丁应用问题	48
JDBC 事件.....	49
MEMORY LEAKS 内存泄漏	50
诊断坏块	51
CORE DUMP.....	53
死锁	54
夯中的 数据库/过程.....	55
实例调整	56
优化器	57
PL/SQL 问题.....	58
恢复	59
自旋进程	60
一些有用的数据库事件.....	61

关于 Oracle 数据库问题诊断信息获取

在对 Oracle 数据库相关情况及问题进行诊断及信息捕获时, 请:

1. 请确认 MAX_DUMP_FILE_SIZE 参数设置, 这将避免用以诊断的追踪文件(trace file)因为空间不足而被截断 !!!
2. 上传 alert.log 文件。11g 中, 文本形式的 alert.log 是放在 ADR home 相应"trace"目录下。而 XML 形式的 alert.log 则是放在 ADR home 下的"alert"目录下。如需验证 ADR home 确切位置, 可以执行"select * from v\$diag_info" 并从其输出中找到。
请看下页诊断追踪文件目录地址信息。
3. 获取 RDA 收集信息 (版本 4.2x 以上更佳). 请查看 MOS 文档 [330363.1](#) 及 [314422.1](#) 以了解更多信息。
4. 请对所遇问题提供详尽描述以便于分析并更快找到问题原因。
对于严重性为 1 级和 2 级事务, 请提供:
 - 相关业务及技术评估
 - 两名 24x7 联系人, 并提供相关人员邮件及电话联系方式。
 - 一名相关管理联系人, 需提供其邮件及电话联系方式。

问题细节描述:

- 受影响日期时间
 - 相关报错号及其文本信息。
 - 相关问题影响 - 数据库实例崩溃, 进程失败等.?
 - 错误出现频率?
 - 错误是否可重演。
 - 错误是否总在特定日期时间发生?
 - 错误发生是否与数据库相关活动有关? 如备份或高负载处理时.
 - 错误是否总是来自于某个应用或某位用户?
 - 第一次产生此错误是在何时? 当时是否正在做何改动?
 - 当时是否有其他错误发生(例如 ORA-7445 或 ORA-600 错误)?
 - 最近系统是否有任何改变?
5. 如果你还不确定 set events 操作会带何种影响, 请不要设置。

6. 随着 Oracle 11g 诊断追踪文件默认地址的改变。下表展示了以前(10g 以及之前版本)和最新 11g(ADR)的跟踪文件目录位置。

诊断追踪文件目录地址

诊断数据文件	之前版本文件路径	ADR 中文件路径
Foreground process traces	USER_DUMP_DEST	\$ADR_HOME/trace
background process traces	BACKGROUND_DUMP_DEST	\$ADR_HOME/trace
Alert log data	BACKGROUND_DUMP_DEST	\$ADR_HOME/alert&trace
Core dumps	CORE_DUMP_DEST	\$ADR_HOME/cdump
Incident Dumps	USER BACKGROUND_DUMP_DEST	\$ADR_HOME/incident/incdir_n

Oracle 数据库 11g 并没有对前台和后台追踪文件做区分。两种类型文件都放在 \$ADR_HOME/trace 目录下。

7. 请注意 BDE 及其开发需要 11g 漏洞 IPS 打包服务. 在 11g 中 IPS(Incident Packaging Service 事件打包服务) 提供了一种对诊断信息打包的简便方法。

所有数据库事件 (严重错误) 的跟踪文件都被存储在 Oracle 自动诊断库(Automatic Diagnostic Repository: ADR)中。其提供的 ADRCI 工具被用于打包所有文件并上传给 Oracle 支持。从 OEM 中也能找到 IPS 相关链接服务。请参考以下文档了解更多相关信息:

[Note:422893.1](#) - 11g Understanding Automatic Diagnostic Repository.

[Note:443529.1](#) - 11g Quick Steps to Package and Send Critical Error

[Note:1091653.1](#) - 11g Quick Steps - How to create an IPS package using Support Workbench

并非 11g 中所有问题都需生成诊断包. 但对于以下相关 11g 错误及 SR 应该考虑获取相关诊断包。

Manageability:

ORA-600 & ORA-7445

ORA-4030 & ORA-4031

DB Admin:

ORA-600, ORA-700 & ORA-7445

HA:

ORA-600 & ORA-7445

ORA-8103 - Object no longer exists
ORA-1410 - Invalid ROWID
ORA-1578 - Oracle data block corruption
ORA-376 - File cannot be read at this time
ORA-353 - There is a log corruption near a block, change, and time

以下是一个例子 →

客户看见 alert.log 中出现以下错误：

```
Errors in file d:\oracle\diag\rdbms\tn11g\tn11g1\trace\tn11g1_ora_201116.trc:  
ORA-00600: internal error code, arguments: [17099], [], [], [], [], [], [], [], [], [], []  
Errors in file d:\oracle\diag\rdbms\tn11g\tn11g1\trace\tn11g1_ora_201116.trc  
(incident=19569):  
ORA-00600: internal error code, arguments: [17099], [], [], [], [], [], [], [], [], [], []  
Incident details in:  
d:\oracle\diag\rdbms\tn11g\tn11g1\incident\incdir_19569\tn11g1_ora_201116_i195  
69.trc
```

...因此我们看见事件号为 19569.

在正确设置 11g 环境变量后，通过 ADRCI (命令) 进入 IPS 打包处理服务:

```
% adrci
```

因为我们已经知道事件号为 19569, 那执行下一步：

```
adrci> ips pack incident 19569
```

...这样就在当前目录下建立了 zip 压缩包.如果你希望压缩包保存在其它目录中, 那么可以在命令后加上路径信息:

```
adrci> ips pack incident 19569 in D:/tmp/whatever
```

此命令会提示信息产生了包含事件号 19569 相关信息的 IPS 打包文件。这个 zip 压缩文件之后需要被上传到 SR 上。

诊断数据库挂起(HANG)事件

请具体表述问题。并询问用户是否能“解冻”数据库?如何做到的?

1) 使用以下命令生成 HANGANALYZE 追踪文件 ==>

```
$ sqlplus "/ as sysdba"
SQL> oradebug hanganalyze 3
... Wait 90 seconds (to 2 minutes) to give time to identify process state changes.
SQL> oradebug hanganalyze 3 (level 4 dumps leaf nodes as well and may be useful, but there is
more impact on the system)
```

使用以下语法在执行簇级 HANGANALYZE:

```
ORADEBUG setmypid
ORADEBUG setinst all
ORADEBUG -g all hanganalyze 3
```

**请注意: Hanganalyze above level 3 仅应该在 Oracle 支持或开发的相关指导下执行。

2) 在运行 Hanganalyze 之后, 登入一个新 SQL 会话来生成系统状态转储(dump)文件==>

```
$ sqlplus "/ as sysdba"
SQL> oradebug setmypid
SQL> oradebug unlimit
SQL> oradebug dump systemstate 266
... wait 90 seconds
SQL> oradebug dump systemstate 266
... wait 90 seconds
SQL> oradebug dump systemstate 266
```

Level 266 和 267 包含了一些堆信息。Level 266 是对单实例的信息转储, 而 Level 267 则是对簇系统 systemstate 转储。请看 [Note:300870.1](#) 对事件 10998 了解更多信息。

****Note:** 我们同样可以通过使用 Hang File Generator (HANGFG) 工具来完成以上 1) 和 2) 中使用 hanganalyze 和 systemstate 收集信息的操作。HANGFG 工具提供了一系列 UNIX shell 脚本命令来自动化收集和生成 hanganalyze 和 sysemstate 追踪文件。在使用 HANGFG 生成并收集追踪文件时需要考虑需要到需要在已处于性能退化的系统中进行诊断的影响。由于影响级别作为此工具参数被传给 HANGFG, 所以用户需要做出决定, 在启用此工具时, 何种级别的影响是可接受的。当用户选择轻度影响或中等影响(选项 1 或 2)作为参数时, HANGFG 工具同样有能力做出调整以适应用户做出的决定。

HANGFG 是 RAC 自识别的, 并且能在 RAC 和非 RAC 环境中运行。请看 [Note:362094.1](#) (the HANGFG User Guide)以了解其相关信息。

3) 当 hang 事件发生后, 获取到的 Statspack 或 AWR/ASH(10g/11g)报告是否具有时效性取决于做 Statspack 快照频繁程度。举个例子, 如果你每隔 1 小时做一次快照, 在下一次快照前发生了 5 分钟的数据库挂起事件。那么快照由于显得时间跨度太长而很难用于分析当时 5 分钟发生的情况。

AWR/ASH 报告上的其他相关信息可以在“数据库性能”章节中找到。

有时在尝试连接数据库时, 你的调试会话也会挂起在那里。这时留给你 3 种可选方法==>

1) 找一个之前已连上的可用会话。

2) 如果你正在使用 10g/11g, 那么你可以使用-prelim 选项来登录数据库:

```
SQL*Plus. Eg:  
sqlplus -prelim / as sysdba  
oradebug setmypid  
oradebug unlimited;  
oradebug dump systemstate 266;
```

3) 使用操作系统调试器来查看运行进程

- 获取服务器 oracle 先关影子进程 ID (请注意并非 oracle 后台进程, 调试后台进程可能导致数据库奔溃)...“ps -ef | grep ora”
- 使用当前系统级调试器(如 dbx, adb, gdb 等)来获取 systemstate:
\$ gdb \$ORACLE_HOME/bin/oracle <process ID>
(gdb) print ksudss(267)

括号中的数字 (267) 是你希望转储 systemstate 的级别。

****对于 RAC 挂起事件请看 “RAC 性能” 部分中对于 racdiag 脚本的使用****

通过以下 systemstate 命令可在同一时点转储所有 RAC 节点状态:

```
SQL> oradebug -g all dump systemstate 267
```

总结, 你可以上传以下文件:

- Hanganalyze 输出
- Systemstate 转储(dumps)
- hangfiles.out (如果使用 HANGFG 工具命令)
- alert.log
- 在一个很短时间内的 Statspack/ AWR/ ASH 报告
- 可用的系统调试监控输出

更多信息, 请查阅以下文档:

- [Note:61552.1](#) - Diagnosing Database Hanging Issues
- [Note:118527.1](#) - How to get a SYSTEMSTATE dump when unable to connect to the database
- [Note:175006.1](#) - Steps to generate HANGANALYZE trace files
- [Note 121779.1](#) - Taking Systemstate Dumps when You cannot Connect to Oracle
- [Note:452358.1](#) - Database Hangs: What to collect for support.
- [Note:362094.1](#) - HANFG User Guide

Process Spins (进程消耗 CPU 资源达 100%)

当进程出现 spinning 现象, 可使用以下命令:

(** Busy spin 术语解释: 由于程序导致进程不断地检查是否条件为 true)

```
sqlplus / as sysdba
SQL> oradebug setospid <OSPID of spinning process>
SQL> oradebug unlimit
SQL> oradebug dump errorstack 3
-- wait for 1 minute
SQL> oradebug dump errorstack 3
```

从 pstack 命令输出中获取进程 call 信息:

```
$ pstack <process id of spinning process>
```

可以使用 Strace 命令来记录系统进程的系统调用及相关进程输入信号等信息:

```
$ strace -p <process id of spinning process>
```

对于 RAC 系统, 可以使用 systemstate 命令转储相关状态, 此命令的使用在 “诊断数据库挂起事件” 部分有提到。

ORA-4031 错误分析

当试图在共享池中分配大块连续内存而失败时, Oracle 会首先从池中清理当前不用的对象从而使空闲内存碎片(chunk:内存块)得以合并。如果这样仍然没有足够大的单个 chunk 来满足分配需要, 则会产生 ORA-04031 报错。有许多 ORA-04031 错误直接原因都是由于共享池的大小或调整不当造成的。

Note:报 ORA-4031 错误的进程并不总是内存消耗的元凶。错误的发生仅是因为此进程无法得到所需内存而造成的。

如果已经按所有步骤正确设置了共享池大小(SHARD_POOL_SIZE), 但此问题仍然产生时, 除了从应用(例如:使用绑定变量查询替代静态 SQL 等)入手进行分析解决问题外, 也可从其他 trace 文件中获得共享池的一些快照信息==>

修改 init.ora 参数文件, 增加以下事件以从追踪文件中获取相关问题信息:

```
event = "4031 trace name errorstack level 3"  
event = "4031 trace name HEAPDUMP level 3"
```

注意: 除非重启实例, 否则这个参数文件设置不会起效。从 Oracle 9.2.0.5 版本起, 除了在请求 heapdump 时使用 level 1,2,3 或 32 你同样可以使用相同等级并加值 536870912.这样将会在此等级上再进一步显示 5 个最大的 subheaps 同时每个 subheap 下显示相关 5 个最大的 heap areas.

如果问题可以重现, 则可在执行有问题的 SQL 语句前, 在会话级别对事件进行设置:

```
SQL> alter session set events '4031 trace name errorstack level 3';  
SQL> alter session set events '4031 trace name HEAPDUMP level 536870914';
```

Level 536870912 转储 5 个最大 subheaps 并且对应每个 subheap 将显示其 5 个最大 heap areas。由于 ORA-04031 错误可能在不同池中发生(共享池, 大池, java 池, 流池等), 其 level 值的设置可参照如下:

Component	Level
PGA	1
SGA	2
Large Pool	32
Streams Pool	64
Java Pool	128

Note: 如果 4031 错误出现频繁, 在实例级设置此事件(heapdump 536870914)将会产生许多大的 trace 文件. 这不仅会影响数据库性能而且可能使数据库挂起(某些情况下可能会使得数据库崩溃). 因此有必要及时使用以下语句关闭此事件追踪:

```
alter system set events '4031 trace name HEAPDUMP off';
```

我们也通过 Library cache 转储来帮助确认产生 ora-4031 问题的游标:

```
sqlplus / as sysdba
SQL> oradebug setmypid
SQL> oradebug unlimit
SQL> oradebug dump library_cache 10
```

请注意:在 Oracle 9.2.0.5+, 10g 和 11g 版本中, 4031 trace 文件默认会在 ORA-4031 发生时产生并存放于 user_dump_dest 目录。如果你的数据库版本是其中的一个, 那么你就不需要进行相关设置来生成 4031 trace 文件。

ORA-4031 诊断 →

- 检查 Alert 日志并查看错误是否记录。注意不是所有 ORA-4031 错误都会记录在 alert 日志中。
- 如果错误被记录, 请检查 SGA 的哪部分收到此错误。是共享池, 大池, java 池或 streams 池?
- 查询 v\$sgastat 以检查是否有组件表现出非正常增长。
- 查询 v\$librarycache 并检查:
 - 有无无效对象 (多为 DDL 语句)
 - 有无重载 (Library cache 可能不够大)
 - 内存命中率 (低命中率可能是非共享游标造成的)
- 检查是否存在高 Version Counts 的游标。可通过 v\$sql_shared_cursor 查询。如果存在某父游标下有许多子游标的情况, 检查不可共享的原因。大量子游标会加快共享池的碎片化。请确认应用正在使用绑定变量方式查询。

更多信息, 请看:

- [NOTE:146599.1](#) - Diagnosing and Resolving Error ORA-4031
- [Note:396940.1](#) - Troubleshooting and Diagnosing ORA-4031 Error
- [Note:554521.1](#) - ORA-4031 QUICK FIX - What to look for and what to do!
- [Note:430473.1](#) - ORA-4031 Common Analysis/Diagnostic Scripts

ORA-4030 报错

这个错误原因是 Oracle 服务器进程不能从操作系统上分配出更多内存。含有 PGA(程序全局区)的进程其内存的分配取决于服务器的设置。对于 dedicated 服务器进程, 其包含了 stack 堆栈和 UGA(用户全局区), 保存有用户会话信息、游标信息和数据分类排序区。在多线程模式配置(shared 服务器)中, UGA 是在 SGA 中进行分配, 所以不对 ORA-4030 报错负责。

因此 ORA-4030 是对需要更多内存 (用于 stack UGA 或 PGA) 来完成工作的进程而言的。

请记住产生 ORA-4030 错误的进程并非总是内存损耗的元凶, 错误的发生仅仅是因为此进程无法取得所需的内存造成的。

当你碰到这个错误时, 意味着你已不能从系统中分配内存。这可能是由于你的进程本身导致的 (如果你的进程正需求很多内存), 也可能是其他原因导致操作系统内存被耗尽 (如过大的 SGA 或者太多进程占用了系统的物理内存和虚拟内存)。如何处理此问题?

1. 确定是否还有足够可用内存? 使用系统专用工具来检查内存使用情况 (如 vmstat, top 等).
2. 是否存在操作系统设定限制? 检查 ulimit 命令检查当前资源限制.
3. 是否存在 Oracle 限制设置? 检查 pga_aggregate_target 和相关会话的 pga 内存值.
4. 找出当前哪个进程正在要求大量内存, 可通过 v\$statname 和 v\$sesstat 以了解当前最忙的进程.

当进程正在稳定增长其所需内存时, 我们可以观察它们的运行:

- 通过 v\$sql_area 可以看到哪些命令正被执行:

```
SQL> select sql_text from v$sqlarea a, v$session s where a.address = s.sql_address and s.sid = <SID>;
```

- 你也可以通过 heapdump 转储来查看它们:

```
SQL> oradebug setorapid 10 (10 是 Oracle 进程 Id, 使用 setospid 以设置当前调试进程)
```

```
SQL> oradebug unlimit
```

```
SQL> oradebug dump heapdump 5
```

- 如果报错问题间歇性发生或者由于进程失败太快以至于难以捕获 (但又很可能是内存大量消耗的根本原因), 则我们可以设置事件来获取 heapdump 转储信息:

```
SQL> alter sssion set events '4030 trace name heapdump level 5';
```

...或者直接在 init.ora 文件中设置 event.

更多其他相关信息请看:

- Note:[233869.1](#) - Diagnosing and Resolving ORA-4030 errors
- Note:[399497.1](#) - FAQ: ORA-4030

ORA-600 错误

理解 ORA-600 错误含义对于评估客户数据库中可能存在的问题非常重要。ORA-600 报错的第一个参数指出了问题所在的相关位置 (如 ORA-600 [XXXX]). 对于此错误:

- 1) 至少需要提供相关的 [trace](#) 文件和 [alert.log](#) 文件.
- 2) 操作系统错误/消息日志 (如果可用的话, 像坏块问题等)

对此问题也有一些新工具可用于加快原因查找。工具使用及其他相关信息支持文档请看:

- [Note:390293.1](#) - Introduction to 600/7445 Internal Error Analysis
- [Note:153788.1](#) - Troubleshoot an ORA-600 Error Using the ORA-600 Argument Lookup Tool

此外, 如果可能, 请设法提供测试用例, 数据库历史改变细节和出错频繁程度等情况。并查找 Oracle 在当时自动转储的任何 system 状态信息文件。

ORA-1801 错误

ORA-1801 报错是指日期格式字符串太长以至无法做转换处理. 这仅发生在当几个长字符串按字面值被转换为一个日期部分时. 如果当前没有相关错误的 trace 文件, 我们可以设置以下事件来获取 errorstack 和 heapdump ==>

对应实例级需在 init.ora 中设置:

```
event="1801 trace name errorstack level 3; name heapdump level 13325"
```

或在系统级设置:

```
alter system set events '1801 trace name errorstack level 3; name heapdump level 13325';
```

或在会话级设置:

```
alter session set events '1801 trace name errorstack level 3; name heapdump level 13325';
```


ORA-7445 错误

当 Oracle 服务器进程做了某些错误的处理(以至于触发了某种形式的信号违背 signal violation)时, 会抛出 ORA-7445 这样一个通用错误, (就像 UNIX 下的 SIGSEGV). 对我们来说, 当碰到 ORA-7445 报错的时候, 最重要的事情就是从 trace 文件中找到 call stack trace 信息, 因为它告诉我们相关违背行为是在哪里发生的。

- 1) 提供相关 trace 文件
- 2) 如果可以的话, 请一并提供相关操作系统错误消息日志
- 3) 如果存在相关 core 文件, 请从其中抓取出 stack trace 信息。

更多相关其它信息请看:

- [Note:1812.1](#) - Ext/Pub TECH Getting a Stack Trace from a UNIX CORE file
- [Note:164968.1](#) - What is an ORA-7445 Error?
- [Note:390293.1](#) - Introduction to 600/7445 Internal Error Analysis
- [Note:211909.1](#) - Customer Introduction to ORA-7445 Errors

另外, 如果可能的话, 请提供如测试用例, 数据库历史改变信息以及错误放生频繁程度信息等相关信息。请查找当时 Oracle 自动生成的 systemstate 转储文件。

其它 ORA-NNNN 报错

这里我们讨论下如何诊断一些未预料的 Oracle 错误(ORA-NNNN)。

在诊断问题前, 做一些例行询问 (按经验总结, 尝试形成一种模式)。例如:

1. 是否总是同一个用户/应用造成的?
2. 是否是在数据导入时发生?
3. 问题是在做了哪些操作或修改了哪些之后产生的?

之后, 我们需要收集更多信息以了解错误是如何产生的, 甚至到哪些语句触发了此问题(如果问题是由一个递归 SQL 语句造成的话)。为了达到此目的我们需要在会话级或实例级建立 Errorstack 事件. 如果你能很容易地重现问题, 那么用会话级, 否则则需要在实例级建立事件。

因此, 假设我们收到一个 ORA-904 "invalid column"报错, 我们可以使用:

```
alter session set events '904 trace name errorstack level 3';  
或  
alter system set events '904 trace name errorstack level 3';  
或  
event = "904 trace name errorstack level 3" (in the init.ora)
```

一旦我们获取了 trace 文件, 我们就能从以下对象中找出实际产生错误的 SQL 语句:

1. "当前执行的 SQL"
2. "当前执行的游标"

然后你就能直接通过这些找到的 SQL 来重现问题。而这已经足以用来确认出原因。

更多相关其他信息请看:

- [Note:37871.1](#) - EVENT: ERRORSTACK - How to get

Database Performance 数据库性能问题

为诊断数据库性能问题所要收集的信息:

- 提供对以下问题的回答 →
 - 性能退化影响到了哪些方面?
 - 是否 DBA 能登陆并检查 v\$视图?
 - 你是否能将性能恢复到正常? 如何做到的?
- 提供 STATSPACK/AWR/ADDM 报告以能对性能问题做一些短期快照. 请不要跑得报告时间段过长超出问题发生时间. 对实例来说, 如果问题持续了半小时, 那么抓取那个时间段中 20 分钟的状态为佳. 如果你能精确把握整个时间过程, 那么在性能问题发生前获取第一个快照, 然后在产生性能问题的过程段中获取另一个快照. 请不要在两个快照间重启实例, 因为这将使得获取的结果变得无意义。
- 你也可以在系统运行得“不错”的时候做一个快照(相同时间长度), 这个快照可以作为和有性能问题的快照对比时基准。
- 如果我们需要一个更短时间片上信息(不如当前当前的分析需要 5 到 10 分钟的详细活动信息), 那么可以查看 ASH (Active Session History 活动的会话历史) 报告输出. ASH 报告工具在判断活动会话数量, 查询会话从事的任务和在一段时间最活跃的 SQL 语句时很有用. 它特别擅长用于分析短暂的性能问题. 运行此报告的 ashrrpt.sql 脚本可以在 \$ORACLE_HOME/rdbms/admin 目录中找到。
- 收集操作系统参数来衡量其活动(在运行正常和性能不佳时):
 - CPU
 - 磁盘 I/O
 - Memory/Swap
 - 进程活动
 - 操作系统监控程序 OS Watcher 输出
- 提供 2 个 Hanganalyze 报告和 3 个 Systemstate 转储信息报告 (具体请看“数据库挂起”部分章节)。
- 提供系统 RDA 收集报告。
- 提供 Oracle LTOM 工具收集的信息。

更多其他相关信息请看:

- [Note:210014.1](#) - How to Log a Good Performance Service Request
- [Note:438452.1](#) - Performance Tools Quick Reference Guide
- [Note:748642.1](#) - What is AWR (Automatic workload repository) and How to generate the AWR report?
- [Note:339834.1](#) - Gathering Information for RDBMS Performance Tuning issue
- [Note:402983.1](#) - Database Performance FAQ
- [Note:164768.1](#) - Diagnosing High CPU Utilization

- [Note:301137.1](#) - OS Watcher User Guide
- [Note:352363.1](#) - LTOM - The On-Board Monitor User Guide
- [Note:391182.1](#) - Database performance competency - Road Map
- [Note:605439.1](#) - Recommendations for Gathering Optimizer Statistics on 10g
- [Note:749227.1](#) - Recommendations for Gathering Optimizer Statistics on 11g
- [Note:465787.1](#) - Managing CBO Stats during an upgrade to 10g or 11g
- [Note:276103.1](#) - Performance Tuning Using 10g Advisors and Manageability Features
- [Note:744963.1](#) - Performance Troubleshooting Guide

监听器挂起及性能问题

当监听器(Listener)发生性能问题时设置一个级别 16 的客户端连接追踪。

在客户端 sqlnet.ora 文件中设置参数：

```
TRACE_LEVEL_CLIENT=16
TRACE_FILE_CLIENT=CLIENT
TRACE_DIRECTORY_CLIENT=full path to directory where you want the trace file created.
TRACE_TIMESTAMP_CLIENT=on
TRACE_UNIQUE_CLIENT=on
```

获取这几分钟监听器进程(tnslnsr)的 truss 命令输出以及(发生问题的时间段)不同时间点上 pstack 信息输出。

=====

Truss ==> `truss -o /tmp/mytruss -faed -p <pid of listener process>`

...这个工具会一直保持追踪直到你终止它。可以使用 `ctrl-c` 来终止 `truss`。如果你是在后台运行此工具命令，那么可以通过 `kill` 来杀掉 `truss` 进程以终止它的运行。

Process stack ==> `pstack <pid of listener process>`

...执行 3 次此命令，之间间隔 30 秒左右

如果可以的话，做一个短暂的监听器追踪:

=====

在服务器端 sqlnet.ora 文件中设置参数：

```
TRACE_LEVEL_SERVER=16
TRACE_DIRECTORY_SERVER=<some_known_directory>
TRACE_FILE_SERVER=server
TRACE_TIMESTAMP_SERVER=ON
TRACE_FILELEN_SERVER=<file_size_in_Kbytes>
TRACE_FILENO_SERVER=<number_of_files>
```

...后 2 个参数是为了避免服务器上的监听器 trace 文件过多而设的。举例如果你在客户端设置了以下参数：

```
TRACE_FILELEN_SERVER=100
TRACE_FILENO_SERVER=2
```

...那么客户端上的 2 个 100K 的 trace 文件会被先后填满。之后客户端会切换回第一个 trace 文件继续写入 trace 信息,覆盖原有的 trace 内容,当写满后,再到第二个 trace 文件中写,如此往复。因此在 trace 信息被覆盖前,你需要将这些文件拷贝到其他目录。

我们也可以通过使用 LSNRCTL 来对监听器进行追踪,并且此工具会一直运行会话结束(除非你主动关闭它)。在 LSNRCTL 命令中不存在“=”符号。

```
LSNRCTL set current_listener listenername  
LSNRCTL SET TRC_LEVEL 16
```

...好了,这样 LSNRCTL 就开始追踪了。让它跑一会,这样我们就能看到追踪到的行为信息以及相关的延迟了。

在跑了至少 5 分钟后,关闭 LSNRCTL:

```
LSNRCTL set current_listener listenername  
LSNRCTL SET TRC_LEVEL OFF
```

在使用 LSNRCTL 进行追踪时,在 listener.ora 文件中设置 ADMIN_RESTRICTIONS_listener_name=ON 以禁止运行时其参数的修改。这样,监听器就会拒绝 SET 命令对其参数的修改了。更多信息请看 [Note:272633.1](#)。

如果需要改变 listener.ora 中包括 ADMIN_RESTRICTIONS_listener_name 本身以及其他参数的设置。我们将不得不手工修改 listener.ora 文件并用 RELOAD 命令使得参数起效。

请注意: 服务器端的追踪并不会在你改了 Oracle Net(服务器端) tracing 参数为 OFF 后关闭。它会随着客户端 tracing 的关闭而自动关闭。如果 tracing 由于疏忽一直保持启用,则它可能导致非常巨大 trace 文件的产生并损耗服务资源。为了正确关闭 Oracle Net server tracing, 在设置 tracing 参数值为 OFF 后主动 stop 并 restart Oracle Net server 进程。这可能会导致一些进程的关闭并引发数据库重启,严重程度取决于当前涉及到的会话数量。

因此,你需要上传以下文件:

- 客户端 sqlnet trace 文件
- 对于客户端低效连接的 Pstack trace
- 监听器进程的 Truss 输出
- 3 个对监听器进程的 Pstack traces (3 of them).
- alert.log
- listener.log
- listener traces

更多其他相关信息请看:

- [Note:214022.1](#) - Oracle Net "Connect Time" Performance Tuning
- [Note:67983.1](#) - Oracle Net Performance Tuning
- [Note:28588.1](#) - Using Truss / Trace on Unix
- [Note:110888.1](#) - How to Trace Unix System Calls

SQL 性能问题

对于 SQL 语句性能问题诊断, 我们需要了解:

- 对于存在性能问题的 SQL 语句. 你能否通过使得性能回归正常? 如何修改?
- 提供 SQL 在良好和较差性能下的执行计划输出.
- 设置级别 12 的事件 10046 追踪以生成 trace 文件(在良好和较差性能情况下) →
`alter session set events '10046 trace name context forever, level 12';`
- 在良好和较差性能情况下的 SQL 运行 Tkprof 输出
- 在良好和较差性能情况下的 SQL 运行时间
- 在执行计划输出中对涉及对象的定义 (如 DDL).
- 在执行计划输出中对涉及基表的行数统计
- 在良好和较差性能情况下的事件 10053 的 trace 输出→

`ALTER SESSION SET EVENTS '10053 trace name context forever, level 1';`

...为了获取有效 10053 trace 信息, 就不可避免需要对 SQL 语句进行硬解析。我们可以通过改变 SQL 语句的大小写或空格多少来达到此目的或通过清空共享池 (在生产环境中并不推荐) 来实现。

对 Oracle 10g 及以上版本:

- 你能使用 AWR/Statspack 报告和 awrsqrpt.sql 脚本来获取 2 个 AWR 快照之间的 SQL 语句执行计划。从 AWR/StatsPack 报告中, 定位产生性能问题的语句的 SQL_ID 并生成以下:
 - 对应 SQL_ID 的 AWR SQL 报告
 - 以 SYSDBA 登陆 SQL*Plus
 - SQL> @?/rdbms/admin/awrsqrpt.sql
 - 请确保所使用 AWR 快照都是来自于获取的 SQL_ID 地方。
- 对 AWR 报告中相应 SQL_ID 的查询提供数据字典信息:
 - 以 SYSDBA 登陆 SQL*Plus.
 - SQL> SELECT PLAN_TABLE_OUTPUT FROM
TABLE(DBMS_XPLAN.DISPLAY_CURSOR('<sqlid>'));

其它信息收集 (特别对于 bug) →

对于执行计划的查询性能问题, 请收集所有相关查询对象输出转储和其统计参数, 通过使用这些信息, 我们可以及时重现这些良好和较差的执行计划。

更多其它相关信息请看→

- [Note:21154.1](#) - EVENT: 10046 "enable SQL statement tracing (including binds/waits)"

- [Note:225598.1](#) - How to Obtain Tracing of Optimizer Computations (EVENT 10053)
- [Note:41634.1](#) - TKProf Simplistic Overview
- [Note:372431.1](#) - TROUBLESHOOTING: Tuning a New Query
- [Note:398838.1](#) - FAQ: Query Tuning Frequently Asked Questions
- [Note:163563.1](#) - TROUBLESHOOTING: Advanced Query Tuning
- [Note:208340.1](#) - Troubleshooting SQL Tuning
- [Note:215187.1](#) - SOLT (SQLTXPLAIN) - Enhanced Explain Plan and related diagnostic info for one DML SQL statement

如果由于数据库升级导致的查询性能退化, 请看 →

- [Note:160089.1](#) - Server Upgrade Results in Slow Query Performance
- [Note 258167.1](#) - Upgrading from 8.1.X to 9.X - Potential Query Tuning Related Issues
- [Note 295819.1](#) - Upgrading from 9i to 10g - Potential Query Tuning Related Issues
- [Note:752662.1](#) - TROUBLESHOOTING Query Tuning
- [Note:748200.1](#) - Slow Performance of Queries After Upgrading To 10g

安装/升级/重连问题

对于数据库在安装或重连时发生问题的信息收集:

- /tmp/OraInstall/oraInstall.err
- /tmp/OraInstall/oraInstall.out
- ../oraInventory/logs/installActions.log
- \$ORACLE_HOME/install/make.log
- 对于补丁安装的 OPATCH 工具日志

更多其他相关信息请看 →

- [Note:181459.1](#) - How to troubleshoot an Oracle database software installation on Unix
- [Note:131321.1](#) - How to re-link Oracle Database software on Unix
- [Note:111796.1](#) - How to resolve Database Creation Assistant (DBCA) errors
- [Note:136818.1](#) - Oracle Server RDBMS Installation Issues
- [Note:168600.1](#) - How to find Server Installation log files
- [Note:262592.1](#) - How to tune your Database after Migration/Upgrade
- [Note:208946.1](#) - Troubleshooting Database Creation
- [Note:208059.1](#) - OS Software Upgrades
- [Note:211559.1](#) - Relinking Guide
- [Note:466181.1](#) - 10g Upgrade Companion
- [Note:601807.1](#) - Oracle 11gR1 Upgrade Companion
- [Note:785351.1](#) - Oracle 11gR2 Upgrade Companion
- [Note:1100303.1](#) - Upgrade to 11g Performance Best Practices

RAC 性能问题

请具体描述问题. 当提到问题发生时间时, 请尽量精确, 如某个特定时间等。

请提供 RAC 配置细节：

- 在集群中节点号
- 确定是否第三方集群件正和集群就绪服务 (CRS) 一起被使用
- 注册盘和表决盘的存储类型
- 确认 ASM 是否正被使用
- 每个节点上数据库实例的数量

RAC 诊断([racdiag.sql](#))脚本被用于提供用户友好向导并对 RAC 挂起会话及低性能情景进行问题排查。脚本包括了用于判断 RAC 挂起原因而进行收集大量重要调试信息的相关操作信息。此脚本在转储 hanganalyze 信息的同时, 并会在本地目录中建立一个 `racdiag<timestamp>.out` 文件。

除了从查询输出中判定锁和等待会话之外, `racdiag` 脚本还会动态生成 `systemstate` 转储和 `hanganalyze` 信息输出。此外, `racdiag` 还提供了内部连接性能的主要度量标准。

从文档 135714.1 中对于 `racdiag` 的描述 → 这个脚本应该能在 RAC 环境中出现会话或系统级挂起时运行。脚本应该以 SYS 用户运行。在运行 `racdiag` 脚本前, `catparr.sql` 应该已经以 SYS 运行过, 以在受影响的数据库中建立相关 GV\$视图。有一些系统挂起类型事件能够阻止 `racdiag` 脚本的运行。如果此脚本跑不起来, 那么建议以 SYS 用户在每个节点获取 `systemstate` 转储信息以帮助调试问题。

请上传以下文件:

- a) `racdiag.sql` 脚本输出文件
- b) 每个节点的 AWR/ASH 报告或 Statspack 输出
- c) 操作系统监控器输出 (如果此监控器正运行在当前系统上)

相关 `racdiag` 脚本及其他信息请看 →

- [Note:135714.1](#) - Script to Collect RAC Diagnostic Information (`racdiag.sql`)
- [Note:412894.1](#) - Diagnosing Hangs and Troubleshooting Performance Issues in a RAC Environment
- [Note:556679.1](#) - Data Gathering for Troubleshooting RAC Issues
- [Note:810394.1](#) - RAC Assurance Support Team: RAC Starter Kit and Best Practices

RAC 实例驱逐问题

请具体描述问题. 当提到问题发生时间时, 请尽量精确, 如某个特定时间等。

请提供 RAC 配置细节：

- 集群中的节点数量
- 是否有第三方集群件正和集群就绪服务 (CRS) 一起被使用
- 注册盘和表决盘的存储类型
- 确认 ASM 是否正被使用
- 每个节点上数据库实例的数量

对于实例驱逐被踢(Eviction)问题的诊断- 如 ORA-29740 错误, 请参考以下文档来做数据收集工作:

- [Note 219361.1](#) - Troubleshooting ORA-29740 in a RAC Environment
- [Note:412884.1](#) - Data Gathering for Instance Evictions in a RAC environment

至少你需要上传:

- a) 对于此类问题, 除了上传数据库实例后台进程相关 trace 文件以外, 也要上传 (每个实例) RAC 特定后台进程 (如 LMON, LMS, LCK, LMD, DIAG 等) 相关 trace 文件.
注意: 如果问题导致实例重起, 请确保以上上传文件来自重起之前的实例。
- b) 每个实例的告警日志文件
- c) 获取被踢的每个实例 AWR/ASH 报告或 Statspack 输出
- d) 使用 diagcollection.pl 工具来收集 Oracle 集群件日志. 更多详情请看文档 330358.1. 运行 diagcollection.pl 时, 使用 '--nocore' 参数选项可降低输出的文件大小。
- e) 系统消息日志, 具体文件位置取决于所使用的操作系统:
 - Linux: /var/log/messages
 - Windows: 使用事件查看器查看保存的应用日志和系统日志 TXT 文件。 - Sun: /var/adm/messages
 - IBM: /bin/errpt -a (被重定向到 messages.out 文件)
 - HP-UX: /var/adm/syslog/syslog.log
 - Tru64: /var/adm/messages
- f) 如果有操作系统监控器运行, 则也请获取所有节点的监控器输出

注意:当 Oracle 实例被启动后, 每个相关后台进程 (PMON , SMON , LMS , DIAG 等) 会产生一个 trace 文件。之后仅当需要转储 trace 文件时, 才会继续对这个 trace 文件进行写入。请不要删除这些文件。如果你删除了属于某个后台进程的 Trace 文件, Oracle trace 信息仍会保持向这个节点位置中写入。这造成了有用的诊断信息被丢失。如果你发现属于某个 Oracle 数据库后台进程 trace 文件被删除, 请按 My Oracle Support 文档中提供的步骤来重建它：

[Note:394891.1](#) - How to recreate background trace file(s) that may have been accidentally deleted

Oracle 集群就绪服务(CRS)问题

对于 CRS 问题:

- 如果正在使用第三方提供的 clusterware (SFRAC, SUN Cluster 等.), 首先判断是否第三方 cluster 正在运行 (参考文档 803661.1).
- 检查当前 CRS 是否正在运行:
 - crsctl check crs
 - crs_stat -t
- 检查是否所有私有内连接都起了并可 ping 通
 - ifconfig -a
 - traceroute <private interconnect IP>
- 检查表决盘 (voting disks) 和注册盘 (OCR) 都可以从所有节点访问到
 - ocrcheck
 - ls -l <full path of voting disk>
- 请提供以下信息:
 - 1) racdiag.sql 输出
 - 2) CLUVFY (Cluster 验证工具) 输出.
 - 3) 系统消息日志. 其存放位置取决于当前操作系统:
 - Linux: /var/log/messages
 - Windows: 使用事件查看器查看保存的应用日志和系统日志 TXT 文件.
 - Sun: /var/adm/messages
 - IBM: /bin/errpt -a (重定向到 messages.out 文件)
 - HP-UX: /var/adm/syslog/syslog.log
 - Tru64: /var/adm/messages 如果监控工具正在运行, 其操作系统监控输出将存放在此目录中
 - 4) 从每个 cluster 节点上的三个目录树中收集所有 CRS 日志文件, trace 文件和 core 转储文件 :
 - \$CRS_HOME (css/log css/init crs/log crs/init evm/log evm/init srvm/log racg/dump log)
 - \$ORACLE_HOME (racg/dump admin/*/hdump)
 - \$ORACLE_BASE (admin/*/hdump)...也可以用另一种方法获取打包的 trace 文件 :
\$CRS_HOME/bin/diagcollect.pl

其他相关信息请看:

- [Note 339939.1](#) - Running Cluster Verification Utility to Diagnose Install Problems

- [Note 272332.1](#) - CRS 10g Diagnostic Collection Guide
- [Note 330358.1](#) - CRS 10gR2/ 11gR1/ 11gR2 Diagnostic Collection Guide
- [Note:289690.1](#) - Data Gathering for Troubleshooting CRS Issues
- [Note:783456.1](#) - CRS Diagnostic Data Gathering: A Summary of Common tools and their Usage
- [Note:556679.1](#) - Data Gathering for Troubleshooting RAC Issues
- [Note:357808.1](#) - Diagnosability for CRS / EVM / RACG as directed by Oracle support

Data Guard 日志传输和物理应用

- 确认归档日志是否成功被传送到备库。
- 判断 Data Guard 设置是非级联设置的还是级联设置, 物理备库设置还是逻辑备库设置。 - 确认主库是归档日志模式并已开启自动归档。
- 确认归档有足够空间。
- 判断是否产生错误的条件都满足。
- 确认备库处于 mounted 状态。
- 确认 managed recovery 正在运行。

其他相关信息请看:

- [Note:237213.1](#) - Troubleshooting Data Guard
- [Note:312434.1](#) - Oracle10g Data Guard SQL Apply Troubleshooting
- [Note:814417.1](#) - Dataguard Information gathering to upload with the Service Requests
- [Note:241374.1](#) - Script to Collect Data Guard Primary Site Diagnostic Information
- [Note:241438.1](#) - Script to Collect Data Guard Physical Standby Diagnostic Information

Data Guard 网络故障

- 检查系统瓶颈: CPU (sar -u), I/O (iostat), Memory (vmstat)
- 数据库等待事件 (ARCH 传输, SYNC 传输等.)
- 查看 Statspack Stats 报告 (重做率"redo rate", 事务率"transaction rate", 平均重做写大小"average redo write size",...)
- 检查网络 (备库是否可连?, 是否有网络错误?, 网络带宽是否足够?)
- 检查 alert.log (主库和备库)文件中的所有错误.
- 检查系统消息日志中的错误. 日志文件位置取决于你所用的系统:
 - Linux: /var/log/messages
 - Windows: 应用日志和系统日志被保存为 .TXT 文件并使用事件查看器查看
 - Sun: /var/adm/messages
 - IBM: /bin/errpt -a (重定向到文件 messages.out)
 - HP-UX: /var/adm/syslog/syslog.log
 - Tru64: /var/adm/messages

更多其他信息请看:

- [Note:241925.1](#) - Troubleshooting 9i Data Guard Network Issues
- [Note:745201.1](#) - Diagnosing Connection Problems with an active Data Guard Broker Configuration
- [Oracle9i Data Guard: Primary Site and Network Configuration Best Practices](#)

Streams 错误及问题

对于问题解决及原因分析, 请→

- 1) 具体描述问题。
- 2) 描述 Streams 设置信息 - local capture or downstream capture, number of Streams database, unidirectional or bi-directional replication, etc.
- 3) 来自源和目标系统的 Alert 文件。
- 4) 当报告 Streams 问题时, 请提供源和目标数据库健康检查报告 (请看 Note 273674.1)
- 5) 运行 Streams 故障发现查询脚本. 参考 Note 729860.1.
- 6) 对于 STREAMS 性能问题:
 - 请参考 Streams Performance recommendations (Note 335516.1).
 - 请参看 Note 290605.1 在源和目标数据库运行 STRMMON.
 - 生成源及目标数据库当时时段 AWR/ASH/ADDM 报告.

更多其他信息请看→

- [Note:746247.1](#) - Troubleshooting Streams Capture when status is Paused For Flow Control
- [Note:335502.1](#) - How To Reinstantiate a Single Table in a Streams Environment
- [Note:749036.1](#) - How to re-synchronize the streams replicated objects online
- [Note:405541.1](#) - Print the LCR's listed in the Error Queue
- [Note:291686.1](#) - LogMiner Utility Release 8.1.x - 10g

Streams 应用(Apply)

当对 Streams Apply 进程进行问题诊断时, 我们需要注意以下 3 种情况:

1) 没有事务到达目的地

- 检查 apply 进程状态并确认其是启用的
- 确认 apply 进程是否还没使事务出队列。
- 检查队列的传播是否有发送事务到应用端
- 检查相关规则是否有在 global, schema 和 table 每个级别上正确建立以控制(获取 capture, 传播 propagation, 应用 apply)
- 检查 init.ora 文件
- 检查 BACKGROUND_DUMP_DEST 下的 trace 文件

2) 事务到达了目的地端, 但没有出队列.

- 确认是否 apply 进程没有让任何事务出队列
- 检查出错队列
- 检查是否有定义冲突的方案
- 检查对象实例化 SCN
- 检查当前数据库 SCN 值
- 检查 BACKGROUND_DUMP_DEST 下的 trace 文件

3) 事务到达目的地端并出队列, 但没有被应用.

- 检查 apply 进程延迟
- 检查出错队列
- 检查是否有定义冲突的方案
- 检查对象实例化 SCN
- 检查当前数据库 SCN 值
- 检查 BACKGROUND_DUMP_DEST 下的 trace 文件

注意: 任何情况下, 请通过 Streams 健康检查脚本以提供报告输出。

请看以下文档以便了解与上文相关的更详细信息 →

- [Note:230898.1](#) - How To Troubleshoot the Streams Apply Process
- [Note:273674.1](#) - Streams Configuration Report and Health Check Script
- [Note:730036.1](#) - Overview for Troubleshooting Streams Performance Issues
- [Note:418755.1](#) - 10gR2 Streams Recommended Configuration
- [Note:789913.1](#) - Streams Troubleshooting guide step by step
- [Note:779801.1](#) - Streams Conflict Resolution

诗檀软件 专业 Oracle 数据库服务 www.parnassusdata.com

Oracle 紧急服务国内热线电话: **400-690-3643**

- [Note:265201.1](#) - Troubleshooting Streams Apply Error ORA-1403, ORA-26787 or ORA-26786

高级队列传播(AQ propagation)问题

如果你碰到高级队列传播问题 (很慢或卡住), 请收集以下信息:

```
sqlplus / as sysdba
```

```
oradebug setospid <spid of the j00 background process executing the schedule>  
oradebug unlimit  
oradebug Event 10046 trace name context forever, level 12
```

--等待 10 分钟

```
oradebug Event 10046 trace name context off  
exit
```

```
sqlplus / as sysdba
```

```
oradebug setospid <spid of the j00 background process executing the schedule>  
oradebug unlimit  
oradebug Event 24040 trace name context forever, level 10
```

--等待 10 分钟

```
oradebug Event 24040 trace name context off  
exit
```

oracledebug tracefile_name 命令能显示 trace 文件在哪里生成

当传播问题发生, 我们需要从所有实例中获取相关 systemstate 转储信息→

```
sqlplus / as sysdba
```

```
oradebug setmypid  
oradebug unlimit  
oradebug -g all dump systemstate 266  
-- 等大约 2 分钟  
oradebug -g all dump systemstate 266
```

相关其他信息请看 →

- [Note:233099.1](#) - Troubleshooting Advanced Queuing Propagation
- [Note:102926.1](#) - Performance Tuning Advanced Queuing Databases and Applications

复制问题(Replication)

相关信息请看:

- [Note:1035874.6](#) - Troubleshooting Guide: Replication Propagation
- [Note:122039.1](#) - Troubleshooting Basics for Advanced Replication
- [Note:231499.1](#) - Multi-Master Replication: Diagnostic Information Required & What To Check

ASM 问题诊断 (包括在空闲空间存在并可用时的 ASM 磁盘空间耗尽 报错)

=====

- 请执行 My Oracle Support [Note:351117.1](#) 中的 ASM 调试脚本 (收集诊断 ASM 空间问题所需信息)

- 提供 ASM alert.log 文件, traces 及操作系统消息文件 (可能同时需要数据库和 ASM alerts 和 traces 文件)

更多其他信息请看→

- [Note:340417.1](#) - Data Gathering for Troubleshooting ASM Issues
- [Note:309815.1](#) - Configuring Oracle ASMLib on Multipath Disks
- [Note:284646.1](#) - Creating and using the kfed utility to view ASM disk header

消息网关(Message Gateway)问题

当消息网关不处理任何信息, 请获取以下诊断信息进行分析:

对 Hang 问题及 系统 State 进行信息收集及转储→

```
sqlplus / as sysdba
oradebug setmypid
oradebug unlimit
oradebug -g all hanganalyze 3;
```

等待 30 秒后再次运行 hanganalyze...

```
oradebug setmypid
oradebug unlimit
oradebug -g all hanganalyze 3;

Sqlplus / as sysdba;
oradebug setmypid;
oradebug unlimit;
oradebug -g all dump systemstate 266;
```

等待 30 秒后再次运行 hanganalyze...

```
oradebug setmypid;
oradebug unlimit;
oradebug -g all dump systemstate 266;
```

当消息网关不处理消息时, 扩大 gateway 日志级别以收集更多 gateway 进程相关信息。

```
sqlplus / as sysdba;
SELECT * FROM MGW_GATEWAY;
exec DBMS_MGWADM.SET_LOG_LEVEL (3);
```

上传 MGW trace 文件.

获取 MGW Java 进程相关 thread 转储文件:

```
ps -ef | grep mgwextproc
```

使用以下脚本命令得到 JVM 下的 thread 转储信息

```
for pids in `ps -ef | grep mgwextproc | awk '{print $2}'`
do
```

```
echo $pids  
kill -3 $pids  
done
```

Trace 文件会在当前工作目录中产生。请上传这些 trace 文件。

诗檀软件 专业 Oracle 数据库服务 www.parnassusdata.com
Oracle 紧急服务国内热线电话: **400-690-3643**

ORA-3113 错误

ORA-3113 错误通常产生自 Oracle 客户端工具。此错误意味着客户端未能和 Oracle 影子进程进行通讯。此错误常常需要我们收集更多的信息来帮助分析原因,且原因也多种多样。因此这种“oracle 影子进程通讯失败”错误是一种“全方位涉及”类型。ORA-3113 常发生在当由于某些原因致 Oracle 服务器进程死亡时。

我们需要更多信息用来判断错误成因:

1. 错误是发生在尝试建立数据库连接时或在一个已经建立的连接上?
2. 在 USER_DUMP_DEST (或 11g 上的 \$ADR_HOME/trace) 目录下存在相关失败会话的 trace 文件。
3. 此错误是因为特定命令造成的或是随机产生的? 如果是由于特定命令造成的, 在会话级别开启 SQL_TRACE 以找到问题命令。

其他相关信息, 请看 →

- [Note:17613.1](#) - ORA-3113 on Unix - What Information to Collect
- [Note:1020463.6](#) - DIAGNOSING ORA-3113 ERRORS

数据库/实例崩溃

为方便解决及原因分析, 请提供以下内容 →

- BACKGROUND_DUMP_DEST 或\$ADR_HOME/trace 下的 Alert.log 文件
- Alert log 文件中最新涉及的错误 (如 ORA-600 或 ORA-7445) 相关的 trace 文件.
- 当时 Oracle 自动收集的`系统状态转储文件`
- RDA (Remote Diagnostic Agent 远程诊断代理)输出, 可参考 Note 314422.1

数据库块讹误(Corruption)

有很多原因可能造成坏块，其包括:

- IO 硬件损坏
- 操作系统问题
- Oracle 造成的问题
- 对数据库中已使用"UNRECOVERABLE"或"NOLOGGING"关键字命令操作的数据恢复（这种情况会提示 ORA-1578 错误）

Oracle 发现并报错的时间可能晚于数据最初产生讹误的时间。

由于并非总能了解当时坏块产生的原因，而多数情况下，关键业务需求数据库立即响应并再次开始运行，因此以下步骤用于处理当前问题：

- 1) 判断数据讹误问题的影响程度及范围，是临时的还是永久不可恢复的错误。如果坏块大范围出现或者错误在各处不断产生，那么就需要先集中精力先确定问题原因(如检查硬件设备等)。这一点很重要，如果你的硬件本身出了问题，那怎么操作都无法恢复系统。
- 2) 替换或移走（可能）存在问题的硬件设备。
- 3) 判断哪些数据库对象收到了影响。
- 4) 选择最恰当的数据库恢复/数据解救方案。

为方便解决及原因分析, 请提供以下内容→

- BACKGROUND_DUMP_DEST 或 \$ADR_HOME/trace 下的 Alert.log 文件
- Alert log 文件中最新涉及的错误（如 ORA-600 或 ORA-7445）相关的 trace 文件
- Block 转储及其他相关诊断信息，可参考 Note 28814.1
- RDA (Remote Diagnostic Agent 远程诊断代理)输出，可参考 Note 314422.1

更多相关信息请看→

- [Note:1088018.1](#) - Master Note for Handling Oracle Database Corruption Issues
- [Note:28814.1](#) - Handling Oracle Block Corruptions in Oracle7/8/8i/9i/10g/11g
- [Note:412566.1](#) - Basics of Debugging/Getting Dumps on Windows and Unix Platforms for Internals/Corruption Analysts
- [Note:76375.1](#) - Prevention, Detection and Repair of Database Corruption

RMAN 报错

问题解决及原因分析→

- [Note:132941.1](#) - RMAN: Quick Debugging Guide
- [Note:815857.1](#) - Troubleshooting RMAN Performance or Hang Issues
- [Note:748257.1](#) - RMAN Troubleshooting Catalog Performance Issues
- [Note:740911.1](#) - RMAN Restore Performance

Oracle 企业管理器 / 网格控制(OEM/Grid Control)

为方便问题解决及原因分析, 请→

- 具体描述问题
- 对于 EM Agent 问题:
 - 打包压缩并上传\$AGENT_HOME/sysman/log 中的文件
 - \$AGENT_HOME/sysman/config/emd.properties 文件
 - [Note 229624.1](#) - How to Log and Trace the EM 10g Management Agents
- 对于 OMS 错误及问题:
 - 打包压缩并上传\$OMS_HOME/sysman/log 中的文件
 - \$AGENT_HOME/sysman/config/emoms.properties 文件
 - [Note 229627.1](#) How to Log and Trace the EM 10g Management Service
 - [Note 421053.1](#)- EMdiagkit Download and Master Index

更多相关信息请看 →

- [Note:311580.1](#) - How to Troubleshoot Metric Collections
- [Note:421053.1](#) - EMdiagkit Download and Master Index
- [Note:421638.1](#) - EMdiagkit Overview

SQL*Net 报错

问题解决及原因分析→

- [Note:16564.1](#) - SQL*Net V2 on Unix - A Quick Guide to Setting Up Client Side Tracing
- [Note:219968.1](#) - SQL*Net, Oracle Net Services - Tracing and Logging at a Glance
- [Note:67983.1](#) - Oracle Net Performance Tuning

补丁应用问题

问题解决及原因分析→

为了调试检查补丁问题, 需要设置:

- export OPATCH_DEBUG=TRUE
- export PERL_DL_DEBUG=1

请看相关信息 →

- [Note:403212.1](#) - Location Of Logs For Opatch And OUI

JDBC 事件

问题解决及原因分析 →

- [Note:338433.1](#) - How to Trace JDBC in 10g
- [Note:434462.1](#) - How To Turn On Server Side Tracing From A JDBC Program

Memory Leaks 内存泄漏

内存泄漏是指当进程始终不能归还其做临时使用而分配的内存时，可用内存在这种分配下逐渐损失的情况。内存泄漏最终可导致可用内存不断变少最终耗尽。如果不做检查，那些正受到内存泄漏的进程将不断增长对内存的需求直到其达到当前系统对用户模式下进程设置所允许的最大值。因此，对内存基线需求和内存泄漏之间的判断就很必要了。当然内存使用的增长并不一定归因于内存泄漏。

问题解决及原因分析→

- [Note:477521.1](#) - How To Troubleshoot Memory Leaks on Microsoft Windows
- [Note:477522.1](#) - How To Troubleshoot Memory Leaks on UNIX
- [Note:1003841.1](#) - Diagnosing swap full problems/possible memory leak issues
- [Note:166490.1](#) - Diagnosing Oracle memory on HP using GLANCE
- [Note:163763.1](#) - Diagnosing Oracle memory on Sun Solaris using PMAP
- [Note:166491.1](#) - Diagnosing Oracle Memory on AIX using SVMON
- [Note:403584.1](#) - Understanding and Diagnosing ORA-600 [729] Space Leak Errors

诊断坏块

1.如果数据库仍然是启动的，那么可以使用下列 sql 来确定在 ORA-1578 或组合 ORA-600 中损坏的对象：

```
select * from sys.dba_extents where file_id = fno  
and bno between block_id and block_id + blocks - 1;
```

注意：oradba 或者 odba 工具可以用来转换 ORA-600 参数到文件和块组件。可以用于上述查询语句。

2.如果是下面之一的对象，那么可以进行重建：

索引

可重建的表

回滚段（除了 SYSTEM 回滚段之外）

排序块（不需要重建）

临时表

3.如果不是上面其中之一，那么下面是一些思路：

-是否运行在 ARCHIVELOG？

-是否有表数据的备份（例如，导出，sqlldr 文本）？

-表上是否有任何 NOT NULL 字段和索引？

-如果是这样，这些索引是否是 unique？

4.如果客户对数据库显示的坏块很担心损坏历史数据，那么可以按照“进一步建议”

5.如果客户运行在归档模式那么告诉他们保留一份归档和在线日志的副本，防止后续诊断的操作。如果不行，至少备份所有的在线日志。（显然，如果在过去的点发生了错误，那么这样做是无意义的，但是你不确定是否值得这样做）

6.建议客户使用 10210,10211 和 10212（极少的）时间来捕获错误的源头。如果你怀疑这不是一个 oracle 错误，那么切换到块检查_db_block_compute_checksums（在 7.2 中是 db_block_checksum）。如果你怀疑块在内存中损坏，那么使用_db_block_cache_protect。注意：这个在大多数平台上没有什么影响，只是一个性能开销

7.在一些场景中，可能需要客户切换到归档模式，来证明或反驳这是 oracle 的问题。

8. 寻找模式。例如：

--所有损坏在相同的设备/控制器上

--所有第 4 个块被损坏

--块是好的，但是被写入错误位置

一部分块是好的 (例如: 头尾不同步)

9. 重建那些有坏块的表, 并且跳过坏块:

A. 使用 10231 事件, 并且在表上执行全表扫描, 注意是否还有坏块, 如果有会报错失败。

B. 使用 hint /*+ rowid(bad_table) */ 来跳过坏块

C. 如果表有唯一索引在非空字段上, 那么久可以手动或 rowid 表 (有所有完好 rowid 的表, 从访问的索引抽取) 来重建表。

D. 其他形式的索引也可以使用, 但可能更难, 并且有丢失好的行数据的危险

10. 使用 10210, 10211 或者 10212 来强制更新这些块软损坏, 并且被发现。然后使用 10231 事件。

11. 在极端情况中 (大表等), 我们可以设置 10231 事件并且在表上重建索引。如果该表通过索引访问, 则会跳过坏块。注意, 这只是一个暂时的解决方案。

Core Dump

1. 确定堆栈和检查顶部的函数（在剥离出错的函数之后，例如 ksedmp）。尝试检查在运行时操作的堆栈。
2. 尽可能的活动 trace 文件。如果没有 trace 文件，那么确定 core 文件中调用的堆栈
3. 确定原因是否可以被复制
4. 如果这个发生在 PL/SQL 对象编译的时候，那么慢慢缩小编译的范围。有些时候观察堆栈可以帮助定位范围。例如其中有一个操作等级函数 evaopn。
5. 如果发生在执行 PL/SQL 对象的时候，那么现在要考虑数据的问题了。

死锁

1. 检查生成的 alert 文件和 trace 文件。可能是由于一个无法访问的文件导致
2. Trace 文件中应该有一个死锁的图表。一个用户进程的进程状态僵死。这个可以给出足够的线索。例如：
 - 1) 如果你看到了 ssx 锁，那么怀疑在丢失了外键索引上进行了级联删除操作
 - 2) 如果你看到了 TX 锁，并且锁模式是 4，那么可以表示在事务表上的竞争，而不是在行级。
3. 询问在应用程序中是否有任何显式的锁定。例如 LOCK TABLE 或者 SELECT FOR UPDATE。这些都是相关的。
4. 应用是否在不同命令中使用了锁
5. 检查参数文件中的 row_locking 和 serializable 参数
6. 如果很好复制，那么使用 leve 4 的 10046 事件来关注所有的死锁会话
7. 如果都失败了，在实例级别设置事件：trace name 60 systemstate 等级为 10

夯中的 数据库/过程

- 1.检查 alter 文件和任何 trace 文件找出证据 (例如 : 'Waited too long for row cache enqueue')
- 2.检查是是否 arch 在归档中出现问题
- 3.将下面查询的结果 spool 到文件 :

```
select * from v$llock where request > 0  
select * from v$session_wait where wait_time = 0;  
select * from v$latchholder;
```

-获得系统状态 (systemstate) 的 dump

(确保 max_dump_file_size 设置的足够大 , 并且确保 user_dump_dest 有足够的空间。如果必要的话 , 使用 oradbx/orambx/oraddebug 来 unlimit trace)

- 4.小心用户对问题的看法。可能仅仅是少了一个分号 , 后台进程的死亡或者不能访问的路径也可能导致
- 5.最根本的原因可能是一个自选过程持有了资源 , 其他事情与之冲突
- 6.在启动时 , 可能是在回滚一个大的终止的事务。

实例调整

- 1.有什么变更？
- 2.确认整个 数据库实例受到了影响
- 3.确认其他非 oracle 的应用（例如 unix 命令），影响不严重
- 4.如果有其他数据库存在，它们是否表现类似的问题
- 5.使用 utlbstat/utlestat 在好的时间点进行数据收集
注意：至少繁忙半小时以上，建议开启 timed_statistics on，如果能收集到数据库正常的 report 的话最好。
- 6.另外值得采样的是 v\$session_wait，因为上述 2 个 sql 可能达不到预期
- 7.如果性能也很糟糕，那么也可以检查下 shared_pool
- 8.不要忘记非 oracle 问题，例如网络，磁盘四度和操作系统问题
- 9.一个很烂的应用可能导致所有数据库变慢
- 10.一旦发现可疑的性能瓶颈，然后尝试解决这个问题并观察结果。试图在一个时间内发生一个变化，观察结果，然后再次循环。

优化器

1. 优化器使用的是:cost 还是 rule ?
2. 如果是 cost , 是否比基于 rule 的更快
3. 有些时候, 客户想影响基于 cost 的优化器选择的访问路径, 但是不要轻易的使用 hint。例如, 包含了临时查询, 第三方工具, 或者在传入内核前某种前缀形式的 sql
这里有一些调整以 cost 为基础的优化器计划的方法列表:
 - 在参数文件中变更 optimizer_mode
 - 创建一个视图并暗示这个。然后用工具从视图中查询
 - 增加或减少 db_file_multiblock_read_count , 来让全表扫描更多或更少。
 - 增加额外的假数据, 来改变谓词的选择性
(并不简单实现, 因为这些假数据需要被过滤)
 - 有选择的性的不分析一些表, 这样就有使其有默认的相关的 cost。这也将有基于 cost 的优化器考虑其不同
4. 在网络上进行查询经常会出现性能问题。例如: 远程表和本地表进行表连接。
5. 进一步分析为什么优化器会选择这个访问路径, 然后设置 10053 事件进行语句解析。
注意, 你还将需要一个索引名称和 objectid 的映射。

PL/SQL 问题

意想不到的行为涵盖的问题太广泛。这个可以缩小为下列类型：

--偶尔会遇到的错误代码

--数据相关

--执行速度很慢

--执行的路径不是开发人员所期望的

1.内核使用的版本是什么？

2.是否有可重复测试的小工具。客户应该提供尽可能小的测试用例，来进行测试。因为客户是熟悉源代码的。

3.当发生了不期望的错误，在整个实例上让用户对每个报错进行 errorstack 事件。这个是直接定位问题代码的第一步

4.使用测试例子进行测试

5.最困难的是将代码分片。如果看到了问题重现。这就可以：

--移除所有过程/函数中的 boby，但是留下调用。

--注释掉从游标中的 feth，但是留下它们的组件

--使用硬编码代替变量（例如参数）

6.添加额外的异常处理过程。在问题的区域，输出判断结果。

7.使用 dbms_output 写出 trace 部分。注意这仅仅表名 PL/SQL 执行到这里，可能帮助很小。

8.将调试信息加入到表中。这有时可能有用，但是与实际不符。

9.写入数据库管道，然后另一会话从中读取，最理想的情况是一个预编译或者 OCI 程序可以把接受的写出到屏幕上，即使一个简单的 PL/SQL 块也可能接受信息，写入表和提交。

恢复

1.最关键的事情可能是完全了解到底发生了什么。这个可以通过：

--和客户沟通（并不总是可信）

--从 alert 日志中

--v\$视图或者 dump 一些事件

2.考虑到最坏情况，在恢复之前，强烈要求客户首先进程备份。

有些情况，很难做到这一点。但是这个是一个明智的举动，它取决于客户是否听从你的建议。

3.要小心客户对于事情的描述。详细 alter 日志

自旋进程

- 1.主要是确定进程在哪里循环。可以使用 orabx,orambx 或者 oradebug。同样的使用操作系统命令也可以检索到相同的信息
- 2.进行堆栈的 dump，将有问题的进程附加上
- 3.检查 v\$session_wait 来观察自旋的会话。一般情况这个是可以确定的，但是有些时候不准确，因为可能做一些形式的工作，例如使用错误的访问路径访问行。

一些有用的数据库事件

所有的事件可以在\$ORACLE_HOME/rdbms/mesg/oraus.msg 中找到信息, 注意这些事件在不同的版本中工作可能发生变化。

- 10015 - dump 回滚段的访问信息
- 10046 - 超级的 sql_trace 设置:
 - Level 4 -sql_trce +绑定变量信息
 - Level 8 -sql_trace +会话等待信息
 - Level 12 - leve4 +leve 8
- 10053 : 转储基于 cost 的优化器如何选择访问路径
 - 注意: 有大量复杂的输出
- 10210 - 执行增加的数据块检查
- 10211 -执行增加的索引块检查
- 10212 -执行增加的簇块检查 d
- 10228 -跟踪 redo 的应用
- 10231 -在全表扫描上跳过硬损坏块

更多资源

技术资源: <http://www.parnassusdata.com/resources/>

技术支持: service@parnassusdata.com

销售: sales@parnassusdata.com

下载 PRM FOR ORACLE 灾难恢复软件: <http://www.parnassusdata.com/>

联系诗檀软件: <http://www.parnassusdata.com/zh-hans/contact>

最后

Parnassus Data

[ParnassusData](#) Corporation, Shanghai, [GaoPing](#) Road No. 733. China

Phone: (+86) 400-690-3643

[ParnassusData.com](http://www.parnassusdata.com)

Facebook: <http://www.facebook.com/parnassusData>

Twitter: <http://twitter.com/ParnassusData>

Weibo: <http://weibo.com/parnassusdata>

Copyright © 2013, [ParnassusData](#) and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, or including implied warranties and conditions of [merchantability](#) or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, [Opteron](#), the AMD logo, and the AMD [Opteron](#) logo are trademarks or registered

诗檀软件 专业 Oracle 数据库服务 www.parnassusdata.com

Oracle 紧急服务国内热线电话: **400-690-3643**

trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0410

Copyright © 2016 ParnassusData Corporation. All Rights Reserved.