

如何解决 ORA-04031 错误

翻译: Fenng

文章内容

- 1.和共享池(shared pool)相关的实例参数
- 2.诊断 ORA-04031 错误
- 3.解决 ORA-04031 错误

- 已知的 Oracle BUG
- 共享池碎片
 - V\$SQLAREA 视图
 - X\$KSMLRU 视图
- 小的共享池尺寸
 - 库高速缓冲(library cache)命中率
 - 共享池大小的计算

- 4.对 ORA-04031 的高级分析

诊断并解决 ORA-04031 错误

对于大多数应用来说,共享池的大小对于 Oracle 性能来说都是很重要的。共享池中保存数据字典高速缓冲和完全解析或编译的的 PL/SQL 块和 SQL 语句。

当我们在共享池中试图分配大片的连续内存失败的时候, Oracle 首先刷新池中当前没使用的所有对象,使空闲内存块合并。如果仍然没有足够大单个的大块内存满足请求,就会产生 ORA-04031 错误。

当这个错误出现的时候你得到的错误信息如下:

```
Error: ORA 4031
```

```
Text: unable to allocate %s bytes of shared memory (%s,%s,%s)
```

```
-----  
Cause: More shared memory is needed than was allocated in the shared pool.
```

```
Action: Either use the dbms_shared_pool package to pin large packages, reduce your use of  
shared memory, or increase the amount of available shared memory by increasing the value of  
the init.ora parameter "shared_pool_size".
```

1.共享池相关的实例参数

在继续之前,理解下面的实例参数是很重要的:

- **SHARED_POOL_SIZE** – 这个参数指定了共享池的大小，单位是字节。可以接受数字值或者数字后面跟上后缀"K" 或 "M" 。"K"代表千字节, "M"代表兆字节。
- **SHARED_POOL_RESERVED_SIZE** – 指定了为共享池内存保留的用于大的连续请求的共享池空间。当共享池碎片强制使 Oracle 查找并释放大块未使用的池来满足当前的请求的时候，这个参数和 **SHARED_POOL_RESERVED_MIN_ALLOC** 参数一起可以用来避免性能下降。

这个参数理想的值应该大到足以满足任何对保留列表中内存的请求扫描而无需从共享池中刷新对象。既然操作系统内存可以限制共享池的大小，一般来说，你应该设定这个参数为 **SHARED_POOL_SIZE** 参数的 10% 大小。

- **SHARED_POOL_RESERVED_MIN_ALLOC** –这个参数的值控制保留内存的分配。如果一个足够尺寸的大块内存存在共享池空闲列表中没能找到，内存就从保留列表中分配一块比这个值大的空间。默认的值对于大多数系统来说都足够了。如果你加大这个值，那么 Oracle 服务器将允许从这个保留列表中更少的分配并且将从共享池列表中请求更多的内存。这个参数在 Oracle 8i 是隐藏的。

2.诊断 ORA-04031 错误

ORA-04031 错误通常是因为库高速缓冲中或共享池保留空间中的碎片。在加大共享池大小的时候考虑调整应用使用共享的 SQL 并且调整如下的参数：

```
SHARED_POOL_SIZE,  
SHARED_POOL_RESERVED_SIZE,  
SHARED_POOL_RESERVED_MIN_ALLOC.
```

首先判定是否 ORA-04031 错误是由共享池保留空间中的库高速缓冲的碎片产生的。提交下的查询：

```
SELECT free_space, avg_free_size, used_space,  
       avg_used_size, request_failures, last_failure_size  
FROM   v$shared_pool_reserved;
```

如果：

```
REQUEST_FAILURES > 0 并且  
LAST_FAILURE_SIZE > SHARED_POOL_RESERVED_MIN_ALLOC
```

那么 ORA-04031 错误就是因为共享池保留空间缺少连续空间所致。

要解决这个问题,可以考虑加大 **SHARED_POOL_RESERVED_MIN_ALLOC** 来降低缓冲进共享池保留空间的对象数目,并增大 **SHARED_POOL_RESERVED_SIZE** 和 **SHARED_POOL_SIZE** 来加大共享池保留空间的可用内存。

如果：

```
REQUEST_FAILURES > 0 并且  
LAST_FAILURE_SIZE < SHARED_POOL_RESERVED_MIN_ALLOC
```

或者

REQUEST_FAILURES 等于 0 并且

LAST_FAILURE_SIZE < SHARED_POOL_RESERVED_MIN_ALLOC

那么是因为在库高速缓冲缺少连续空间导致 ORA-04031 错误。

第一步应该考虑降低 SHARED_POOL_RESERVED_MIN_ALLOC 以放入更多的对象到共享池保留空间中并且加大 SHARED_POOL_SIZE。

3.解决 ORA-04031 错误

- ORACLE BUG

要解决这个错误,进行的诊断的第一步是在你的平台上使用最新的补丁集。大多数的 ORA-04031 错误都和 BUG 相关,可以通过使用这些补丁来避免。下面表中总结和这个错误相关的最常见的 BUG,可能的环境和修补这个问题的补丁。

BUG	描述	Workaround	Fixed
<Bug:1397603>	ORA-4031 / SGA memory leak of PERMANENT memory occurs for buffer handles.	_db_handles_cached = 0	8172, 901
<Bug:1640583>	ORA-4031 due to leak / cache buffer chain contention from AND-EQUAL access	Not available	8171, 901
<Bug:1318267>	INSERT AS SELECT statements may not be shared when they should be if TIMED_STATISTICS. It can lead to ORA-4031	_SQLEXEC_PROGRESSION_COST=0	8171, 8200
<Bug:1193003>	Cursors may not be shared in 8.1 when they should be	Not available	8162, 8170, 901

共享池结构中的一些 BUG 会引起这个错误,不过通常大量的共享的 SQL/PLSQL 语句也会引起这个错误。一旦打过了最新的补丁,在遇到这个问题的时候我们强烈推荐调整数据库和应用。

要得到已知的 BUG 的完整信息,可以参考: <Note:62143.1>: Main issues affecting the Shared Pool on Oracle 7, Oracle8 and Oracle8i.

- 共享池碎片

每一次,需要被执行的 SQL 或者 PL/SQL 语句的解析形式载入共享池中都需要一块特定的连续的空间。数据库要扫描的第一个资源就是共享池中的空闲可用内存。一旦空闲内存耗尽,数据库要查找一块已经分配但还没使用的内存准备重用。如果这样的确切尺寸的大块内存不可用,就继续按照如下标准寻找:

- ◇ 大块(chunk)大小比请求的大小大
- ◇ 空间是连续的
- ◇ 大块内存是可用的(而不是正在使用的)

这样大块的内存被分开，剩余的添加到相应的空闲空间列表中。当数据库以这种方式操作一段时间之后，共享池结构就会出现碎片。

当共享池存在碎片的问题,分配一片空闲的空间就会花费更多的时间,数据库性能也会下降(整个操作的过程中, "chunk allocation"被一个叫做"shared pool latch" 的门所控制) 或者是出现 ORA-04031 错误 errors (在数据库不能找到一个连续的空闲内存块的时候)。

参考 <Note:61623.1>: 可以得到关于共享池碎片的详细讨论。

如果 SHARED_POOL_SIZE 足够大，大多数的 ORA-04031 错误都是由共享池中的动态 SQL 碎片导致的。可能的原因如下：

- ◇ 非共享的 SQL
- ◇ 生成不必要的解析调用 (软解析)
- ◇ 没有使用绑定变量

要减少碎片的产生你需要确定是前面描述的几种可能的因素。可以采取如下的一些方法，当然不只局限于这几种：应用调整，数据库调整或者实例参数调整。

请参考 <Note:62143.1>，描述了所有的这些细节内容。这个注释还包括了共享池如何工作的细节。

下面的视图有助于你标明共享池中非共享的 SQL/PLSQL:

•V\$SQLAREA 视图

这个视图保存了在数据库中执行的 SQL 语句和 PL/SQL 块的信息。下面的 SQL 语句可以显示给你带有 literal 的语句或者是带有绑定变量的语句：

```
SELECT substr(sql_text,1,40) "SQL", count(*) , sum(executions) "TotExecs"
FROM v$sqlarea
WHERE executions < 5
      GROUP BY substr(sql_text,1,40)
      HAVING count(*) > 30
      ORDER BY 2;
```

注意：语句 Having 中的 "30"数值可以根据需要调整以得到更为详细的信息。

• X\$KSMLRU 视图

有一个固定表 `x$ksmlru` 跟踪共享池中导致其它对象换出(age out)的应用。这个固定表可以用来标记是什么导致了大的应用。

如果很多对象在共享池中都被阶段性的刷新可能导致响应时间问题并且有可能在对象重载入共享池中的时候导致库高速缓冲竞争问题。

关于这个 `x$ksmlru` 表的一个不寻常的地方就是如果有人从表中选取内容这个表的内容就会被擦除。这样这个固定表只存储曾经发生的最大的分配。这个值在选择后被重新设定这样接下来的大的分配可以被标记，即使它们不如先前的分配过的大。因为这样的重置，在查询提交后的结果不可以再次得到，从表中的输出的结果应该小心的保存。

监视这个固定表运行如下操作：

```
SELECT * FROM X$KSMLRU WHERE ksmlrsiz > 0;
```

在 Oracle8i 中这个表不能被 SYS 用户之外的用户所选取。

• 小的共享池尺寸

最后,一个小的共享池可以导致 ORA-04031 错误,不过在碎片真正的是个问题的时候增大共享池的大小的时候要小心。在错误发现的时候通常有延迟现象,不过当在大的共享池的碎片中找到一片空闲的内存会加大对性能的影响。

下面的信息将有助于你调整共享池的大小：

库高速缓冲命中率

命中率有助于你衡量共享池的使用，基于多少次 SQL/PLSQL 需要被解析而不是重用。下面的 SQL 语句有助于你计算库高速缓冲的命中率：

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

如果 misses 比上 executions 大于 1%，那就应该尝试着通过加大共享池来减少库高速缓冲的丢失。

Shared Pool Size Calculation

要计算最适合当前工作负荷的共享池大小，参考：

<Note:1012046.6>: HOW TO CALCULATE YOUR SHARED POOL SIZE.

4.对 ORA-04031 的高级分析

如果使用如上的解决办法，这个错误仍然出现，在 `initSID.ora` 文件中设定如下的事件并重新启动实例：

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

会在下一次错误发生的时候产生一个跟踪文件。

这个跟踪文件可以提供给 Oracle 支持人员来解决问题。

相关文档

[<Note:151790.1>](#) : Oracle8 Tuning Documentation Guide

[<Note:62143.1>](#): Understanding and Tuning the Shared Pool in Oracle7, Oracle8, and Oracle8i.

[<Note:1012046.6>](#): HOW TO CALCULATE YOUR SHARED POOL SIZE

[<Note:1012049.6>](#): TUNING LIBRARY CACHE LATCH CONTENTION

[<Note:61623.1>](#): Resolving Shared Pool Fragmentation In Oracle7

[<Note: 146599.1>](#)：就是这篇文档的英文原稿。

About:

这篇文章的翻译是本着学习交流的目的,当然,原英文作者保留版权.如果对译文上的技术细节不是很满意,可以参考原英文文档.有其它问题请通过 Fenng@itpub.net 和我联系.