

Oracle 数据库回滚段的故障分析与性能优化

Fault Analysis and Performance Optimizing on Oracle Rollback Segment

梁毅冬 (中国工商银行软件开发中心 510630)

摘要:回滚段是 Oracle 数据库的重要组成部分,数据库的性能与回滚段息息相关。本文将针对数据库回滚段常见的故障进行系统的分析,结合实例提出相应的解决方案,并对数据库回滚段的改善与优化进行了探讨。

关键词:Oracle 回滚段 故障分析 优化

1 回滚段概述

为了更好地理解回滚段故障的成因,我们有必要深入了解回滚段的结构和工作方式。

回滚段为数据库提供了事务回滚、数据库恢复及保证数据读一致性。每个 Oracle 数据库实例 (Instance) 都会包含至少一个回滚段,不管事务是否被提交,被事务改变了的数据的旧值都将记录在回滚段中。

回滚段的工作方式可以简单概括为自动扩展、循环分配,后文将结合回滚段故障分析对此做更详细的说明。需要指出的是,回滚段与事务之间是一对多的关系,即一个回滚段可以处理多个事务,但每个事务且仅由一个回滚段负责处理,不能扩展到别的回滚段中,直到事务结束。

2 回滚段常见故障分析

如果回滚段出现故障,轻则使数据处理无法执行、用户无法读写数据,重则导致数据库不能正常启动或关闭,导致数据库瘫痪。若能将回滚段的故障排除,一般说来是不会影响用户的实际数据。实际上绝大多数的回滚段故障都是由于数据库管理员对回滚段的认识不足所造成。较为常见的错误有以下几种:

(1) 快照过期;

(2) 回滚段空间分配问题;

(3) 数据库不能正常启动或关闭。下面将分别进行分析,并给出相应的解决步骤。

2.1 快照过期 ORA-1555: Snapshot too old

这个错误代码指出问题是出在快照过期,是一个涉及到数据读一致性的问题。通常这种错误多出于

数据库在执行一个长时间的事务或查询的同时有一些小的在线事务被提交的情形中。这种错误与回滚段的工作方式密切相关。前面提到了回滚段的工作方式是自动扩展、循环分配。

当一个事务使用回滚段时,发现当前的区段 (extent) 已满,它将在同一回滚段中寻找一个可用的区段来继续事务。Oracle 会首先检查回滚段的下一个区段是否被一个活动的事务占用,如果没有,则 Oracle 将其分配给当前需要回滚段的事务。事务在提交时会告诉 Oracle 可以释放它所占用的回滚段区段,但为了保证数据读一致性,Oracle 不会马上执行,这一区段中的信息将至少保留到在事务提交前就开始执行的查询操作结束后才释放。问题的关键就在于回滚段的空间不是无限延伸的,它在用户定义的范围内是有限的,当回滚段中最后一块区段使用完后,Oracle 将不得不循环使用其在本回滚段中第一次定义的区段块。如图 1 所示,当事务 T1、T2 所使用的 E3 区段占满后,系统将自动分配 E4 给 T1、T2 使用,在 E4 写满后,T1、T2 将重新使用 E1。

笔者在使用 Oracle 的时候就曾经遇到过这种问题,由于要生成一份跨部门的数据统计报表,需要运行一个涉及对多个数据表的查询,这个过程耗时很长。在开始查询之后,另外一个部门的人员对其中一个查询所涉及到的表 A 进行了更新 (T1),这是部门之间缺少沟通而经常出现的情况,由于更新的数据不多,T1 很快提交。此时 T1 所使用的回滚段区段被置为非活动的,但里面的数据依然保留,以保证读一致性。紧接着又提交了一个有大量更新操作的事务 T2,它的操作与

前面查询所涉及的表无关,但由于需要大量回滚段,根据回滚段循环使用的法则,被置为非活动的、包含了 T1 更改的数据旧映像的区段将被覆盖,以提供给 T2 使用。而就在此时,根据读一致性的要求,前面查询需要使用 T1 更改前的表 A 的数据,但发现在回滚段中这些数据已经被 T2 所覆盖,无法再获取了。查询被迫异常终止。这时 Oracle 就提示 ORA1555 错误,指出快照太旧或回滚段太小。

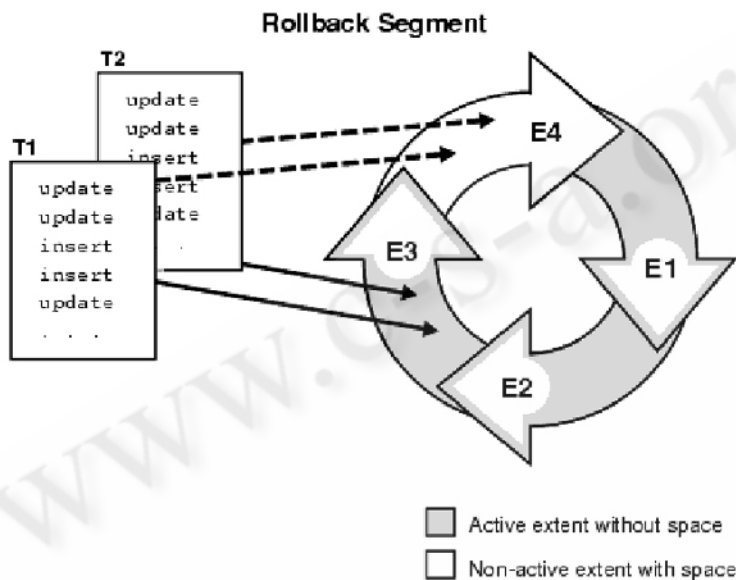


图 1

解决方法是增大回滚段的大小,以此来降低类似上述情形发生的概率,同时也应该注意,运行批量和进行耗时很长的作业应该尽可能的安排在非繁忙时间上进行,这样就可以有效地减少 ORA1555 错误的发生。

2.2 回滚段空间分配问题

这类错误中最常见的是 ORA - 01650 : unable to extend rollback segment A by NUM in tablespace B。表示回滚段 A 所在的表空间 B 不足。在进行一个需要大量回滚段的数据操作时,有可能导致现有回滚段不足,但可用于分配的回滚段表空间已满,无法再分配更多的空间。问题可以通过适当增加回滚段空间的大小来解决。ORA - 01562: failed to extend rollback segment NUM 也是类似的错误,只不过在指出出错回滚段时使用的是回滚段 ID 号而不是回滚段的名称而已。

另一个与回滚段空间分配有关的错误是 ORA - 01628: max # of extents num reached for rollback seg-

ment num。当一个已经达到最大区段的回滚段企图继续分配新的区段的时候,这个错误就会发生。我们可以增大回滚段的 MAXEXTENTS 或 PCTINCREASE 的值,在数据字典中找出缺少空间的对象,修改其 STORAGE 参数值。SQL 命令为 ALTER TABLESPACE XXX STORAGE (MAXEXTENTS number) 来增加 MAXEXTENTS, 其中 number 值必须大于错误信息中所指的数值,但注意不能大于 LARGEST MAXEXTENT 的值。

2.3 数据库不能正常启动或关闭

这种错误比较复杂,也最不容易定位,因为导致数据库不能正常启动或关闭的原因很多。首先要找出问题出在哪里,是否与回滚段有关。有时候回滚段的故障会连锁引起其他的错误,Oracle 在给出错误提示的时候,并没有直接指出是回滚段错误,仅仅从表面现象是很难判断的,我们可以通过查看 SIDALRT. LOG 文件来获取这方面的信息。警告日志可以在 init.ora 文件中的 background_dump_dest 参数所指的目录下找到。一般来说,如果数据库不能正常启动或关闭是由于回滚段引起,那么这个引起故障的回滚段多半是有物理故障的,而导致故障的前导因素可能是上文提到的几种故障没有得到及时的解决,也可能是包含回滚段的数据文件的物理损坏。在这里我将结合本部门的一个数据库系统发生的故障及解决过程进行说明。我们使用的是 Oracle8.1.6 的版本,在运行中,数据库出现了挂起的现象。在 infoALRT. log 文件中(应用的 ORACLE_SID 为 info)发现有以下日志记录:

```
ORA - 00603: ORACLE server session terminated by fatal error
```

```
ORA - 00600: internal error code, arguments: [4152], [], [], [], [], [], [], []
```

```
ORA - 01562: failed to extend rollback segment number 2
```

```
ORA - 01650: unable to extend rollback segment RBSA1 by 128 in tablespace RBSA
```

这里 Oracle 进程是被异常终止的。ORA - 01562: 和 ORA - 01650 错误在上文已经解释过了,从这段日志中可以看出问题是出在一个叫 RBSA1 的回滚段上,这个回滚段是在 RBSA 表空间里的。这时数据库的进程

监控 (PMON) 检测到错误并开始执行进程恢复。系统监控进程 (SMON) 也参与了数据库的恢复。我们可以通过 infoPMON. TRC 和 infoSMON. TRC 查看到 PMON 和 SMON 的执行 TRACE。

```
ORA - 00600: internal error code, arguments:
[4136], [0], [], [], [], [], [], []
```

```
ORACLE Instance info (pid = 6) - Error 600 encountered while recovering transaction (2, 49).
```

显然, Oracle 在对事务进行恢复的时候出现了问题。这里 PID=6 是指 SMON。因为数据库挂起, 我们试图使用 shutdown immediate 来关闭数据库, 但数据库关不下来, 只有强制关掉数据库。

由于我们使用的数据库是非归档方式, 回滚段的数据文件的备份也有问题, 经过分析, 回滚段出问题, 数据库正在将数据汇总到一份汇总表中, 这项汇总操作我们可以重新运行, 不会影响其他事务数据, 于是决定丢失回滚段中的那一事务。但回滚段中几乎可以肯定包含着活动的事务。因此, 坏的那个数据文件不能脱机 (offline) 或是 drop 掉。

处理办法是首先关掉数据库, 编辑 initINFO. ora 文件。在 initINFO. ora 文件中加入如下一行: `_corrupted_rollback_segments = (RBSA1)`, 这里的 RBSA1 就是之前出问题的回滚段。这个参数的作用是使指定的回滚段失效, 它只能在指定的情况下才使用。在控台上以 restricted 模式打开数据库: STARTUP RESTRICT, 然后删除掉出错文件 RBSA. ORA 所属的回滚段表空间, 即 RBSA1。DROP TABLESPACE RBSA1 INCLUDING CONTENTS; 在操作系统中将 RBSA. ORA 移去, RBSA. ORA 在系统中的路径可以在 infoALRT. log 文件的出错日志中找到, 这里是 "J: \ORANT \DATABASE \SYSTEM \". 接着重建回滚段表空间 RBSA,

```
CREATE TABLESPACE RBS DATAFILE 'J: \ORANT \DATABASE \SYSTEM \RBSA. ORA'
```

```
SIZE 2048M AUTOEXTEND OFF MINIMUM EXTENT 8M
DEFAULT STORAGE ( INITIAL 8M NEXT 8M MINEXTENTS 8 MAXEXTENTS UNLIMITED );
```

```
重新在回滚段表空间 RBSA 上建立回滚段 RBSA1,
CREATE ROLLBACK SEGMENT RBSA1 TABLESPACE RBSA
```

```
STORAGE ( INITIAL 8M NEXT 8M OPTIMAL 80M
```

```
MINEXTENTS 5 MAXEXTENTS UNLIMITED );
```

```
创建完后使回滚段 online,
```

```
ALTER ROLLBACK SEGMENT RBSA1 ONLINE;
```

```
然后 ALTER SYSTEM DISABLE RESTRICTED SESSION;
```

使普通用户可以使用该数据库。

最后我们在 initINFO. ora 中把重新创建的回滚段 RBSA1 包括进来, 并且移除掉 `_corrupted_rollback_segments` 这个参数。以便下次启动数据库的时候自动将 RBSA1 回滚段 ONLINE。

但在某些情况下, 我们是不希望丢失回滚段中的那些事务。可以先将损坏的那个数据文件从最近的备份中恢复, 然后 MOUNT 上数据库, 注意这里不是 OPEN。查看数据文件是否联机, 如果不是, 必需先把它联上。通过下面的 SQL 语句列出所有的联机的重做日志和他们的序号及首次改变号 (first change numbers) :

```
select v1. group #, member, sequence #, first_change#
```

```
from v $ log v1, v $ logfile v2
```

```
where v1. group# = v2. group#;
```

对非归档模式的数据库还要执行如下的查询: SELECT FILE#, CHANGE# FROM V \$ RECOVER_FILE; 将查询出来的 CHANGE# 值与上文的 FIRST_CHANGE# 的最小值进行比较, 如果 CHANGE# 大, 就可以用联机日志进行恢复, 如果小, 则数据库是不能用联机日志恢复的。用 RECOVER DATAFILE '`<full_path_file_name>`' 进行恢复。如果恢复成功, 将返回 Media recovery complete。这个时候就可以 OPEN 数据库了。

与数据库回滚段有关的错误远不止这些, 这里只是简要地对几类经常出现的错误进行了说明。我们可以通过查看回滚段的状态判断回滚段的工作是否正常。SELECT SEGMENT_NAME, TABLESPACE_NAME, STATUS FROM DBA_ROLLBACK_SEGS; 就可以把回滚段当前状态和所属的表空间都列出来。ONLINE 是正常工作的状态, OFFLINE 也属于正常, 但只是处于非工作状态, 而 NEEDS RECOVERY、PARTLY AVAILABLE 和 INVALID 则表明回滚段是有问题的, 这需要特别关注了。我们可以通过 ORACLE 的 LOG 文件, 分析原因, 再具体提出解决方案。

3 性能优化

数据库处理事务的能力与回滚段密切相关,它在数据库中起着关键性的作用,如果它设计得不合理,将会严重影响系统的性能。在 ORACLE 数据库中,我们可以通过以下视图 V \$ ROLLNAME、V \$ ROLLSTAT、V \$ UNDOSTAT、V \$ SESSION、V \$ TRANSACTION 来观察数据库回滚段的性能。在建立和调整优化数据库回滚段时,应该注意以下几个方面:

3.1 为回滚段单独分配表空间

这样做的好处是为了减少系统的 I/O 争用。因为数据库系统实际上还是建立在操作系统之上的,ORACLE 的表空间与操作系统的文件存在映射的关系。回滚段的作用决定了回滚段的写入与数据和索引的写入是并行进行的。为回滚段创建单独的表空间,就是为了让回滚段的数据文件与记录数据字典、用户数据、索引等的数据库文件分离开来。如果可能,我们建议回滚段的数据文件应该和其他数据库文件放置在不同的磁盘上,这样能各自充分发挥硬件的作用,将 I/O 争用降到最少,同时也使得数据库管理变得更容易。此外,我们知道表空间里的碎片会严重影响数据库的性能,特别是在进行查询操作的时候,影响会更加突出。而回滚段是需要经常性收缩的,这使得表空间的自由块更容易形成碎片。分离回滚段可以减少数据库表空间的碎片产生。因此,为回滚段单独分配表空间是十分必要的。

3.2 创建不同大小的回滚段

在本文前面已经提到过,每个事务只能使用一个回滚段,因此,回滚段应该充分大,以容纳一个完整的事物处理。一般说来,小的事务在一个拥有众多小回滚段的数据库中有较好的性能,而对一些需要长时间运行的事务,比如说批量作业,则是在有较大回滚段的数据库中表现出色。所以,对于小的事务,我们应该建立小的回滚段,这如果可以使回滚段缓存在内存或 SGA 区中,那么数据库将因为磁盘 I/O 的减少而使性能得到提高。对于大的事务,通过被预先分配在一个足够大的回滚段中,减少了由于回滚段数据块扩充导致的性能下降,而从中获益,同时也降低了上文提到的出现回滚段空间不足错误的可能。但是,对大多数数据库系统而言,是既包括小的事务,也包含大的批量,这就要求我们去创建不同大小的回滚段,为不同类型

的事务提供服务。这并不是说要为每一个事务都显式地指定一个回滚段,这样做会加大数据库系统管理的复杂度,但对一些特殊的事物,比如大的批量作业,显式指定一个大的回滚段还是很有必要的。我们可以通过使用 SET TRANSACTION USE ROLLBACK SEGMENT a_big_rs 来实现。据 ORACLE 方面的统计,大多数的 SQL 语句只会影响所涉及表的 10% 以内的数据行,将每个回滚段的大小设置为数据库中最大表大小的 1/10 是比较恰当的,这样可以为大多数的 SQL 操作提供充足的空间。至于建多少个回滚段才算是合适,要视具体情况而定,一般建议是以并发的事务数来衡量,为每四个并发事务提供一个回滚段。

3.3 设置回滚段配额

我们知道大的和不正确写的事务都会消耗很多资源。利用回滚段配额,对用户进行分组,然后给组分配最大能使用的回滚空间。当到达了组的限制后,将不能产生任何新的事务,直到当前事务的回滚空间被释放。在 ORACLE 中可以使用参数 UNDO_POOL 来定义多少回滚空间能被一个资源组使用。通过给不同的用户群分配不同的回滚段配额可以有效地管理数据库系统的回滚段,对不同的用户区别对待,实现资源的有效利用,从而实现数据库性能的优化。

4 总结

回滚段在数据库系统中起了很重要的作用。作为一名 DBA,不仅要正确认识和理解回滚段的工作原理,还要知道如何采取合适的步骤去处理回滚段的错误。对数据库回滚段的优化配置不是一次就能完成的,必须对回滚段进行长期而细致的观察,根据实际情况做出调整,合理分配回滚段的使用,使数据库系统的整体性能得到提高。

参考文献

- 1 Kevin Loney, George Koch 著,钟鸣、石永平等译, Oracle 9i 参考手册,机械工业出版社,2003-7。
- 2 Abraham Silberschatz, Henry F. Korth, S. Sudarshan. Database System Concepts, Fourth Edition. McGraw-Hill Companies, Inc, 2002.
- 3 Oracle Server Administrator's Guide. Oracle Corporation, 2001.