

基于 Oracle9i 关系数据库性能优化策略

The Research of the Optimized Strategy based on Oracle9i

熊曾刚 (孝感学院计算机科学系 432100、武汉湖北大学数学与计算机科学学院 430062)

张学敏 (孝感学院计算机科学系 432100、武汉理工大学计算机学院 430063)

王时绘 (武汉湖北大学数学与计算机科学学院 430062)

摘要: 本文从 Oracle9i 的系统结构着手,分析了可能影响 Oracle9i 数据库性能的几个因素,并对数据库服务器、数据库系统全局区 SGA 的参数、数据库回滚段、磁盘 I/O、SQL 语句、数据库碎片、数据库系统应用程序等几个方面的性能给出了优化的策略。

关键词: Oracle9i 数据库 性能 优化 策略

1 引言

一个成功的数据库应用系统要经过需求分析、概念设计、逻辑设计、物理设计、程序设计、数据库实施等多个阶段。其中物理设计阶段中数据库性能的优化好坏会影响到整个应用系统能否正常运转。因为在大型数据库应用系统中,用户的数据存放量大、访问量也大,系统的效率和性能是非常重要的,而实际应用中,由于数据库性能的调整不完善或者设计不合理,则可能导致数据库系统的高效运行。因此,我们在数据库应用系统设计和运行过程中,要根据实际情况随时对系统性能进行优化和调整,使应用系统高效运行。

Oracle9i 是一个优秀的大型关系型数据库系统。它能为一个规模很大,用户数很多的系统提供高性能的服务。Oracle9i 数据库性能优化的调整是一项系统工程,涉及到的方面很多。数据库管理员需要对数据库运行状况进行综合评估,以判定数据库运行性能的好坏。本文从几个方面分别论述了如何对

ORACLE9I 关系数据库进行性能优化,以使数据库应用系统能高效地运行。

2 影响 Oracle9i 数据库性能的因素

在 Oracle9i 数据库系统中,主要有以下几个因素影响 Oracle9i 数据库的性能:

(1) 数据库服务器性能。数据库服务器是整个系统的核心,它的性能高低直接影响到整个系统性能,尤其是随着 ERP 系统的流行,很多应用软件是外购的,用户根本没有机会从应用软件的角度出发去调整性能,只能在服务器运行环境方面进行系统性能优化。

(2) 数据库的参数调整。这里涉及到影响系统全局区 SGA 大小的几个参数,通过对它们的监控分析来有效提高数据库系统对内存的利用率,同时还应兼顾系统平台资源(如:操作系统、后台用户进程)对内存的需求。另外还有一些其他参数(回滚段、排序等)也需要从应用出发合理配置。

(3) 数据库物理文件的合理分布及逻辑对象(表空间、表、索引)的合理分布,这将影响数据库的读写速度。数据库应用中有大量的 I/O 操作,如何减少数据库 I/O 访问量、提高 I/O 操作速度对提高数据库性能非常关键,我们可以通过对系统 I/O 的监控、对应用访问的分析,合理的将表、索引等放在不同的表空间,将数据库物理文件分布在不同的磁盘上来提高数据库访问速度。

(4) 用户进程如何合理利用数据库的资源。这包括应用程序如何用正确的方式编写 SQL 语句、如何有效使用索引、如何合理的将进程分布在服务器或客户端上,从而提高对数据库 SGA 内存区的命中率、减少 I/O 访问、减少对网络带宽占用等。

(5) 数据库碎片的管理。数据库碎片对数据库的影响主要表现在以下三方面:一是降低性能。过多的碎片导致在数据访问时需要更多的磁盘 I/O,增加了数据库读的数量和磁盘查找时间,从而引起性能下降。二是影响系统的稳定性。碎片会严重影响数据库系统

的稳定性,如应用在创建新的存储对象或分配新的存储空间时因找不到足够大的连续空间而终止。三是浪费存储空间,碎片带来很多无效的空间使用,导致这些空间不能存储实际数据。

3 Oracle9i 性能优化的主要策略

在Oracle9i数据库中,系统性能优化的策略主要可集中在两方面进行:一方面是在系统设计和开发阶段对库结构和应用程序代码进行优化,如采用方法库、合理使用非3NF、合理使用索引、合理安排分布业务处理过程;采用完整性约束、存储过程、数据库触发器;改写效率低下的SQL语句,优化事务处理过程,尽量少占用ORACLE资源;另一方面,可在应用系统运行阶段由DBA对系统参数进行调整,达到性能优化的目的。

下面我们就以上提到的五个影响Oracle9i性能的因素提出主要的优化策略。

3.1 调整Oracle9i数据库服务器性能的优化策略

Oracle9i数据库服务器性能很大程度上依赖于运行的服务器的操作系统,如果操作系统不能提供最好性能,那么无论如何调整,Oracle9i数据库服务器也无法发挥其应有的性能。对操作系统的调整可从内存及硬件高速缓存、硬盘的I/O利用和分配、后台进程的优先级及调度三方面考虑。

在数据库系统中如果占用98%的内存和占65%的内存所获得的性能没有大的差别时,此时应把多的内存让出来,以便更好的完成其他任务,同时也会降低数据在操作系统的页面交换次数。

在数据库系统中,所有后台和前台数据库服务器进程应设置成相同的优先级。

3.2 调整数据库参数的优化策略

3.2.1 调整系统全局区SGA的参数的优化策略

Oracle9i数据库启动的各种环境参数都保存在参数文件INIT.ORA中,这些初始化参数对SGA的大小具有决定性影响。参数

DB_BLOCK_BUFFERS(SGA中存储区高速缓存区数目)是SGA大小和数据库性能的最重要的决定因素,该值较高,可以提高系统的命中率,减少I/O。每个缓冲区的大小等于参数DB_BLOCK_SIZE的大小。参数SHARED_POOL_SIZE(分配给共享SQL区的字节数),是SGA大小的主要影响者。

(1) 调整数据库缓冲区高速缓存。从内存读取数据比从硬盘读取数据要快得多。因此Oracle9i总是把最近经常使用的数据块调入数据缓存。该部分的命中率决定了数据库的性能。数据缓存的使用情况记录在V\$SYSSTAT中,可通过下列SQL命令:

```
Select name, value from v$sysstat
Where name in('dbblock gets','consistent gets','physical reads');
```

由此可知数据缓存的命中率为: $1 - \frac{\text{physical reds}}{\text{dbblock gets} + \text{consistent gets}}$

如果命中率在60%-70%之间,则应增加数据缓存的容量,可以增大DB_Cache_Size的值。

(2) 调整LIBRARY CACHE。库高速缓存中包含私用和共享SQL区和PL/SQL区。调整SGA的重要问题是确保库高速缓存足够大,以使ORACLE能在共享池中保持分析和执行语句,提高语句分析和执行效率,降低资源消耗。可以使用SQL命令:

```
Select SUM(pins),SUM(reloads) FROM
V$librarycache;
```

如果 $\frac{\text{SUM(reload)}}{\text{SUM(pins)}} \approx 0$, 说明LibraryCache的命中率比较合适,若大于1,则需要增加SHARED_POOL_SIZE的大小。

(3) 调整数据字典高速缓存。数据字典高速缓存包括了有关数据库结构、用户实体信息等,数据字典的命中率可以用以下SQL语句实现:

```
Select (1 - (SUM(getmisses) / (SUM(gets) + SUM(getmisses)))) * 100 FROM V$rowcache;
```

如果其值大于90%,说明命中率较好,否则应增加SHARED_POOL_SIZE的大小。

(4) 调整REDO缓存。REDO缓存存放了日志信息。这对数据库的维护和恢复有很重要的影响。我们可以用以下SQL语句查看REDO缓存的空间请求次数:

```
SQL name, value FROM v$sysstat WHERE
name='redo log space requests';
```

如果以上结果为0,则比较好,若此值不断增大,则需增加LOG_BUFFER的值。

3.2.2 调整回滚段优化策略

回滚段是一个存储区域,数据库使用该存储区域存放曾经一个事务更新或删除的行的原始数据值。回滚段控制着数据库处理事务的能力,因此回滚段的设计对Oracle9i的性能优化显得非常重要。回滚段的设计应遵守的原则是:

(1) 分离回滚段。分离回滚段是指单独为回滚段创建一个或多个表空间,使回滚段与用户数据、索引等分离开来,由于回滚段中的写入与数据和索引是并行进行的。将回滚段单独分离开来可以减少I/O争用。独立回滚段可使数据库管理变得方便,可以节约磁盘空间。回滚段经常变化,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碎片的产生。

(2) 依不同情形创建不同大小的回滚段。对于一些联机事物处理,会对少量数据不断进行修改,此时需创建一些较小的回滚段。

而对一些较大的事物,我们需要建立较大的回滚段。并且尽可能地将这些回滚段分散到不同的表空间上。我们也可以使用如下SQL命令来判定回滚段的数据:

```
Select class, count FROM V$waitstat
WHERE class IN('system undo header',
System undo block', 'undo header', 'undo
block');
```

```
Select sum(value) FROM V$sysstat WHERE
name IN('dbblock gets','consistent gets);
```

如果任何一个class/sum(value) > 10%,那么考虑增加回滚段。回滚段的数量一般按如下规律设定(见表1)。

表1 回滚段数量与并发事物关系

并发事物(n)	回滚段数
$N < 6$	4
$16 \leq n < 32$	8
$n \geq 32$	$N/4$

3.3 调整磁盘 I/O 优化策略

磁盘的I/O的速度对整个Oracle9i数据库系统的性能有较大的影响。影响磁盘I/O性能的主要原因有磁盘I/O竞争、I/O次数过多和数据块空间的分配管理。

对于使用频率较高的物理文件,可以采用以下优化策略:

(1) 将组成同一个表空间的数据文件平均分配在尽可能多的磁盘上,做到磁盘之间I/O负载均衡;

(2) 为表和索引建立不同的表空间,并尽可能将表数据和索引数据存放在不同磁盘上,减少数据文件和索引数据文件对磁盘的竞争;

(3) 将数据文件与重做日志文件分离在不同的磁盘上,减少数据文件和日志文件对磁盘的竞争;

(4) 尽可能将两个或多个日志文件放在不同的磁盘上,减少日志文件对磁盘的竞争;

(5) 要对不同有应用系统创建不同的表空间,尽可能把表空间对应的数据文件存放在不同的磁盘上,减少应用系统之间对磁盘的竞争。

3.4 调整 SQL 语句的优化策略

SQL语句在编写过程中的运用可以大大优化数据库的性能。在SQL语句编写过程中一般应遵循以下原则:

(1) 用NOT EXIT代替NOT IN;

使用NOT EXISTS代替NOT IN会使查询添加限制条件,由此减少必需的全表扫描次数。

(2) 使用带有提示的NOT EXISTS或NOT IN代替MINUS

MINUS将从一个查询中返回一组行,这一组行不会出现在第二个查询返回的行中。使用NOT EXISTS或NOT IN 重写查询将允许它们利用索引,减少子句所需的全表扫描的次数。

(3) 使用UNION ALL代替UNION;

UNION要求进行排序操作来消除两组行中所有重复的行,而UNION ALL会返回所有行(即使有些行是重复的)。如果重复行不重要,则使用UNION ALL 可以避免潜在的大量消耗资源的排序、合并和过滤操作。

(4) 使用WHERE代替HAVING;

HAVING子句仅在聚集GROUP BY子句收集的行之后才对这些行加以限制。如果可能,最好在合并检索的行并将其排序到总计中之前,限制检索的行数。在将行添加到总计中之前,使用WHERE而非HAVING将清除这些行。

注意:HAVING子句应用于聚集函数时不能由WHERE代替。

(5) 避免在WHERE子句中对索引字段施加函数;

因为在WHERE子句中对索引字段施加函数,将导致索引不能用。

另外,在SQL语句运用中,要充分利用ORACLE9I提供的触发器、存储过程等机制来不断的优化数据库的性能。尽可能地把数据库的一些操作放在数据库服务上进行,这样可以减少网络传输带来的瓶颈。避免将数据库服务器的数据通过网络传到客户机,然后在客户机上进行一系列的操作之后再通过网络传送到数据库服务器上。这样只会使数据库性能大大降低。

3.5 管理数据库碎片的优化策略

数据库碎片分表空间级、表级和索引级三种类型,不同类型的碎片产生原因不同,处理方法也不尽相同。

(1) 表空间碎片是由于段的建立、扩展和删除所引起的。所以表空间的碎片需要在数据库结构设计时进行良好的规划。清除表

空间的方法是:

① 是重组表空间;

② 通过Alter Tablespace TablespaceName Coalesces语句消除表空间碎片。

③ 利用EXPORT和IMPORT实用程序,使用先导出数据再导入数据库的方法消除表空间碎片;

(2) 表碎片是指行迁移和行链接产生的ORACLE数据块不连续而形成的碎片。对表碎片的优化策略是:

① 合理设置数据块的大小,使其有足够的空间存储每一行的记录。当然也不能设置太大,太大也会引起系统总的开销,而且数据块的大小只能在建立数据库时进行设置。

② 设置PCTFREE和PCTUSED参数。对时常删除的表PCTUSED较高, PCTFREE较低;对时常删除的表PCTUSED较低, PCTFREE较高。

③ 在表中尽量避免变长记录。

(3) 索引碎片主要是由于索引键太少、索引树失衡和叶节点排序混乱而引起的,索引碎片问题不如表空间碎片和表碎片问题那么严重。对其优化策略是:

① 在主键(Primarykey)的索引方面,不应有超过25%的列成为主键,而只有很少的普通列,这会浪费索引空间;

② 在索引的使用效率方面,当选择数据少于全表的20%,并且表的大小超过Oracle9i的5个数据块时,使用索引才会有效,否则用于索引的I/O加上用于数据的I/O就会大于做一次全表扫描的I/O;

③ 使用索引尤其应当注意的是,在表连接操作时的驱动表/被驱动表的关系。Oracle9i核心使用至底向上、从右至左的规则,如:FROM子句中的最后一个表才是Oracle9i用做为驱动表的表;WHERE子句的最后一个条件中所含的列,它所属的表才是最先被引用的表。总之,在FROM子句中,将表名按被驱动表——驱动表排序,在

Where子句中,将条件语句按最少约束——最多约束排序。

3.6 调整系统应用程序的优化策略

在Oracle9i的OLTP环境中,一般采用Client/Server方式,如何减少Client与Server之间的网络I/O,是整个系统性能提高的重要环节。

减少网络I/O的优化策略是:

(1) 建立与客户端无关的数据库的完整性约束,只在服务器中运行,从而有效减少网络的I/O;

(2) 对于表与表之间较为复杂的事务处理,采用触发器;

(3) 为经常使用到的查询列建立索引,提高系统的查询性能,减少磁盘的I/O操作;

(4) 复杂的查询建立视图,使CLIENT端的程序开发应用简洁;

(5) 多表的连接操作应该从返回较少行上驱动;

(6) 充分利用ORACLE的PROCEDURE和PACKAGE,把有关的过程、函数或过程函数的程序包放在SERVER端,应用程序只是简单调用存储过程和函数,这样可以减少大量的网络I/O,同时对应用程序的维护提供方便。

4 结束语

Oracle9i是一种应用广泛的面向对象的关系型数据库系统。在实际应用环境中,我们应

根据实际情况选择较合理的数据库系统设计和优化策略,这样才能充分利用ORACLE9i提供的高性能服务,使应用系统能够充分发挥其高效的功能。当然,每一种ORACLE性能的优化策略的实施总是相互制约的,所以我们在实际应用过程中,必须全面考虑,综合权衡,这样才能更好地发挥Oracle9i的效益。

参考文献

- 1 飞思科技产品研发中心, ORACLE9i企业管理器详解[M], 电子工业出版社, 2002。
- 2 飞思科技产品研发中心, ORACLE9i数据库高级管理[M], 电子工业出版社, 2002。
- 3 赵松涛, ORACLE9i中文版入门与提高[M], 人民邮电出版社, 2002。
- 4 张曜等, ORACLE9i中文版基础教程[M], 清华大学出版社, 2002。
- 5 于雪平, ORACLE 数据碎片管理[J] P125-128 计算机应用, 第 22 卷第 8 期。
- 6 苏大成等, 基于 ORACLE 数据库开发系统的物理设计优化策略[J] P115-116 第 28 卷第 2 期。
- 7 何泽恒等, ORACLE 应用系统的性能参数分析与优化[J] P303-306, 第 18 卷第 3 期。