

# 基于 Oracle9i 的数据库性能优化策略研究

## The Research of the Optimized Strategy Based on Oracle9i

王振铎 (西京学院 经济系 陕西 西安 710123)

王振辉 (西北工业大学 软件学院 陕西 西安 710065)

**摘要:**本文从 Oracle9i 的系统体系结构入手,分析了影响 Oracle9i 数据库性能的主要因素,主要从软件部分进行优化,操作系统、数据库结构规划、数据库表空间、回滚段、临时表空间、内存、磁盘 I/O、SQL 语句,应用程序,存储过程等多个方面给出了具体的优化策略。

**关键词:**Oracle9i 数据库 策略 内存 磁盘 I/O

### 1 引言

Oracle9i 数据库结构先进但是也十分复杂,它的复杂性允许用户做细小的调整来提高整体性能。但性能调整的方法很多,数据库管理员需要对数据库运行状况进行综合评估,以判断数据运行性能的好坏。本文从几个方面分别论述了如何对 Oracle9i 关系型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 影响数据库性能的因素分析

#### 2.1 合理组织数据库结构,减少 I/O 和资源竞争

数据库中有大量的 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、不同应用,不同数据用途(如系统数据与用户数据),合理地将表、索引等对象放在不同的表空间,将数据库物理文件分布到不同的磁盘上来提高数据库访问速度。

#### 2.2 设计合理的数据表结构,提高存取及检索速度

理论上合理的数据库结构应该是在满足需求的前提下,尽量减少数据的重复量,尽量按关系数据库理论方法设计数据库,以保证数据的完整性和一致性。但是从优化数据库性能的角度来讲,往往故意增加一些冗余数据,以避免多表连接查询操作造成的响应时间缓慢的问题。比如:冗余一个姓名字段,对在存有大量历史记录航空乘客查询系统中查询乘客的信息时,速度的优势将十分明显。同样的,对经常进行查询的表使用索引可以减少磁盘 I/O 操作,提高查询的性能。

对经常一起使用、需要连接的表,建立聚簇可以显著提高性能。

#### 2.3 数据库参数的调整

这里涉及到影响系统全局区 SGA 大小的几个参数,比如:可以通过对增加 Redo\_logbuffer 参数的大小来加快日志恢复的速度,通过对它们的监控分析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、后台进程)对内存的需求。另外还有其他一些参数(回滚段,排序、索引类型)也需要从应用出发合理配置。

#### 2.4 应用程序的优化

数据库性能的好坏,用户一般会直接从应用程序与数据库交互的过程判断,直接的因素就是数据返回或存取的时间。影响数据响应的快慢,与程序中嵌入的 SQL 语句的执行速度,有效数据的网络传输数量有关,另外一个不可回避的问题就是应用开发人员程序中未考虑到 Oracle 数据库的特性,由于 Oracle 提供了一些独立于标准 SQL 语法的谓词和专用函数,采用 Oracle 独有的特性一方面提高了速度和性能,另一方面简化了应用的开发时间。比如:Oracle 的树型查询语句,对查询层次数据十分高效。

#### 2.5 操作系统的优化

操作系统是数据库管理系统表演运行的直接舞台,它能够提供给数据库管理系统多少资源,或者其自身减少资源的占用,都会直接给数据库的性能带来益处。尤其对于外包开发的软件,用户无法从应用软件

的角度去调整性能,只能对运行环境方面进行性能的优化。比如:一台专用数据库服务器比带有 Web 服务的操作系统的数据库服务器,性能要高很多。

### 3 Oracle9i 性能优化的主要策略

对数据库系统的优化,包含两方面的含义,一是在数据库设计时从逻辑结构及物理结构上优化其设计,使之在满足需求的情况下,保持最佳的时间和空间的开销,二是在运行时尽量采用优化的执行策略,选择一条最佳路径来执行 SQL 语句,任何 DML 语句,如 SELECT, INSERT, UPDATE 和 DELETE 的处理的优化是系统优化重要的一部分。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 合理组织数据库结构,减少 I/O 和资源竞争

通过合理地组织数据库的逻辑及物理结构,包括位置、容量、数量,来提高数据库的性能。主要原理是由于磁盘 I/O 是系统性能的瓶颈,解决好磁盘 I/O 的问题,就可以显著提高性能。多个进程同时对同一磁盘进行存取会产生磁盘竞争。通过查询 V \$ FILESTAT 可以知道每个物理文件的使用频率。

```
SELECT name, phyrds, phywrt FROM v $ datafile
df, v $ filestat fs WHERE df. file# = fs. file#;
```

其中, phyrds, 每个数据库文件读的次数; phywrt, 每个数据库文件写的次数。

对于使用频率较高的物理文件可以使用以下原则解决:

- (1) 系统表空间与用户表空间单独存放在不同的磁盘上。
- (2) 将数据文件与 REDO LOG 文件分离在不同的磁盘上。
- (3) 将共享基础信息表数据与其它业务数据放在不同的磁盘上。
- (4) 不要使用超过 70% 的磁盘空间,剩余的磁盘空间存放系统临时数据和作为磁盘碎片整理程序存放中间数据。
- (5) 数据表空间系统默认太小,造成动态扩充,性能下降,对大应用建议采用 1G 以上系统临时表空间和回滚段表空间增大 40M,并且增加几个回滚段。

#### 3.2 设计合理的数据表结构,提高存取及检索速度

- (1) 适度冗余。减少数据库冗余的设计思想产生

于 70 年代,它是促使 DBMS 进步的重要动力之一。然而,以目前国内计算机市场价格一个 120G 硬盘 800 元,减少冗余节约存储空间已不重要。目前已进入软件主导的时代,因此,最容易理解,开发最快,维护最简单,查询统计性能最快的数据库结构才是最好的。在保证数据完整性,一致性不受威胁,有些冗余不足为虑。

(2) 利用索引。索引在逻辑上和物理上都独立于数据,所以,索引可以随时建立和删除,而不会影响表或其它的索引。如果一个索引被删除,所有的应用程序仍能继续运行,但是存取与本索引有关的数据时会降低速度。

(3) 利用聚簇。聚簇是存储表数据的一个可选方法。聚簇是一组经常在一个或者多个表中被使用的公共列数据,获者是多个应用程序共享的列数据。把这些相关的数据存储在磁盘上,可以减少存取磁盘的时间。聚簇能提高数据的检索性能,尤其是进行多表连接时,这一点就更为突出。

#### 3.3 数据库参数的调整

(1) 系统全局区参数调整。系统全局区 (SGA 区) 是 Oracle9i 分配的共享内存区,在这个存储区内存放一个 Oracle9i 例程的控制信息和数据。SGA 区中的数据供同时连接到数据库的所有用户共享。为了优化系统的性能,减少磁盘的输入输出操作,SGA 区的容量应尽量大,以存放更多的数据。

通过调整数据库参数文件和 SGA 区中各个结构的大小,可极大地提高系统性能。

(2) 数据块空间使用控制。ORACLE 的数据存储在数据块中,数据块对应一定的磁盘物理存储空间,数据块大小是在建立数据库时设定的。数据块中包含空闲空间可用于在插入新行、修改数据行时提供附加空间。这个空间的大小由空间管理参数 PCTFREE 和 PCTUSED 控制。如我们设定 PCTFREE = 20,表明在块中要保留百分之二十的自由空间,用于对现行块做更新操作。这时,对块中的行数据插入最多只能达到块空间的百分之八十。当空闲空间达到设定的参数值时,则认为数据块已满,不能进行插入操作。PCTUSED 参数设定块空间被使用的百分比。当被使用的空间小于 PCTUSED 时,允许对块做插入新行操作。当被使用的空间大于或等于 PCTUSED 时,只能对现有行数据进行

更新操作。

一般 PCTFREE 和 PCTUSED 之和要小于或等于 100, 具体要在空间使用效率与处理代价和 I/O 性能之间作出折中。一般按实践值取 CTFREE 和 PCTUSED 之和为 90。

(3) 判断回退段数量是否满足系统运行的要求。回滚段是用于存放一个事务更新或删除行的原始数据, 回滚段太小将不能完成大的事务处理和恢复。通过下面的步骤来判断回退段数量是否能满足系统运行需要

• 查询 V \$ WAITSTAT 表

```
SELECT class, count FROM V $ WAITSTAT WHERE
class IN ('system ndo header', 'system undo block', 'undo header', 'undo block');
```

• 查询 V \$ SYSSTAT 表

```
SELECT SUM ( value ) FROM V $ SYSSTAT WHERE
name IN ('db block gets', 'consistent gets');
```

• 判断回退段数量是否足够

如果任何一个 class/sum ( value ) > 10%, 那么考虑增加回退段。回退段数量一般规律是根据用户来设置:

用户数回退段数量

$n < 16$  4

$16 < n < 32$  8

$32 < = n$   $n/4$  不超过 50

回退段的大小还要根据事务的大小来分配合适的回退段:

长事务要分配大的回退段; 联机事务处理一分配小而多的回退段, 一般来说回退段是系统自行分配的, 但根据事务的特性, 可用 SET TRANSACTION 命令为该事务指明使用的回退段。

### 3.4 应用程序的优化

在 Oracle9i 应用过程中, 一般采用 Client/Server 或 Browse/Server 方式, 如何减少网络 I/O, 是提高系统性能的重要环节。

在应用程序中对 SQL 语句优化, 主要是减少 SQL 参与执行的数据量, 用高效的谓词和句法减少 SQL 的编译时间。对 SQL 语句的优化方法一般是这样的: 利用在表上建立的索引和把经常使用在一起的查询的表形成簇, 对 SQL 语句进行等价变换(改变 SQL 语句执行

顺序, 简化 SQL 语句的操作), 如此来提高 SQL 语句的执行效率, 缩短执行时间。具体方法策略如下:

(1) 优化 NOT 子句。如果在 SQL 语句的 WHERE 子句中有某一列名条件表达式含有不相等子句(! = 或 NOT =), 则对该条件表达式不可以使用索引, 当 NOT 与其它操作结合在一起时, ORACLE 可把它们转换成一个可用索引的操作, 从而提高执行效率。可进行的转换操作如下:

NOT > 转化为 < =

NOT > = 转化为 <

NOT < 转化为 > =

NOT < = 转化为 >

(2) 优化 OR 子句

一般来说, 在 OR 子句中的所有列如果都有索引则进行优化, 如果某些列有索引, 而另一些列无索引, 则可能不做优化。

例如, 查询 SELECT \* FROM EMP WHERE DEPTNO = 10 OR JOB = 'MANAGER';

若在列 DEPTNO 与列 JOB 上都有索引, 那么, ORACLE 将使用这两个索引, 而且该查询等价于以下两个查询的并。

[查询一]

```
SELECT * FROM EMP WHERE DEPTNO = 10
```

[查询二]

```
SELECT * FROM EMP WHERE JOB = 'MANAGER'
AND DEPTNO != 10
```

这样, 在 WHERE 子句中把具有查询速度最快的索引的列名条件表达式放在第一个, 把扫描中会扫过最多记录的行的列名条件表达式放在最后面, 使得对“不等于”的检验次数最少。

(3) 优化 GROUP BY 子句

首先来看两个查询例子

[查询一]

```
SELECT JOB, AVG ( SAL ) FROM EMP WHERE JOB!
= 'PRESIDENT' AND JOB! = 'MANAGER' GROUP BY
JOB
```

[查询二]

```
SELECT JOB, AVG ( SAL ) FROM EMP GROUP BY JOB
HAVING JOB! = 'PRESIDENT' AND JOB! = 'MANAGER'
```

查询一的速度远比查询二的查询速度要快得多。原因是:HAVING 子句完成的功能与 WHERE 子句的功能一样,但两者不等价。HAVING 子句无法取代 WHERE 子句。查询二表中的所有行都被处理,而有些行可以 WHERE 子句“丢掉”。

#### (4) 优化连接

表连接查询的使用是比较频繁的。因为表连接查询涉及到多个表,而表与表之间是笛卡尔积的关系,所以,往往会由于选择查询条件不当,使得大量的时间是在做空操作,浪费时间,效率低下。

优化连接的一般方法:出现在连接子句中的列最好有索引;书写 SQL 语句时尽量使索引可用。对于仅有一个索引可用的情况,通常用其他表作为驱动表。如 EMP 表中 Deptno 列建立索引而 Dept 表中 Deptno 列没有建索引,则下面查询以 Dept 表作为驱动表;

```
SELECT Ename, Dname FROM Dept, Emp WHERE
EMP. Deptno = Dept. Deptno AND JOB1 = Salesman
```

#### (5) 用 NOT EXIST 代替 NOT IN

使用 NOT EXIST 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。

##### · 使用存储过程或函数

使用过程化结构可以减少应用访问 Oracle9i 的次数,降低网络流量。例如执行 10 个单独的 SQL 语句,需要 10 次访问 Oracle9i,但是如果把他们作成一个过程或函数,则只需要访问一次。多个用户使用同一 SQL 语句时,只做一次语法分析只在编译时进行语法分析,运行时不再重做。一来节省内存,二来多个用户可以共享。

### 3.5 操作系统的优化

Oracle9i 数据服务器性能很大程度上依赖与运行的服务器的操作系统,如果操作系统不能提供最优化性能,那么无论如何调整,Oracle9i 也不能发挥其应有的新能。对操作系统的优化可以从以下几个方面考虑。

#### 3.5.1 在专用服务器上运行 Oracle9i

Oracle9i 是内存消耗大户,不要在执行下列功能的系统上运行 Oracle 数据库:

- (1) 主域或备份域控制器 (Windows 2000 下都称域控制器)
- (2) 文件服务器
- (3) 打印服务器

#### (4) 远程访问服务器

#### (5) 路由器、代理或防火墙

### 3.5.2 去除一些不必要的功能和服务

不要使用花哨的壁纸(如果使用,要尽量减小壁纸文件大小)和屏保程序。

#### (1) 禁止非必须的服务;

最好禁止系统里非必须的服务,如果时而需要某些服务功能,可将启动类型设置为“手动”,要做到这一点首先同网络管理员验证实际的服务需求,例如你的系统不需要打印机,通常停止这个服务并设置为手动,停止 License Logging Service 服务除非你对它有特殊要求,不应该使用 DHCP 服务,并禁止它;

#### (2) 不要自动启动你不需要的程序;

(3) 检查菜单“开始/程序/启动”里的内容,删除不需要的程序。

(4) 优化网络配置,网络配置是性能调整的一项很重要的内容,而且很容易隐藏性能瓶颈。配置网卡使用最快速度和有效模式 大多数缺省安装是 NIC,如果这是可选的就尽量调整为“全双工”和最大化线速度,同时删除不需要的网络协议,只保留 TCP/IP 协议。

#### (5) 为 Oracle 禁止或优化文件共享

理想地应该禁止文件共享功能来最小化安全泄露和网络交通,但如果你需要使用文件和打印共享,那么就配置系统中每个网卡的“连接属性”,设置“最大化网络应用程序数据吞吐量”。

## 4 结束语

本文所列举的数据库优化策略是作者工作经验的总结,实际上,数据库的优化是一个长期的比较分析和调整的过程,因为数据和应用都在不断的变化发展中。

### 参考资料

- 1 史森、夏丽丽等译,Oracle 数据库开发指南,清华大学出版社,1998。
- 2 宋瀚涛、李新社,数据库编程与应用,电子工业出版社,1998。
- 3 范剑波、张晓云,网络数据库技术与应用,西安电子科技大学出版社,2004。