

# ORACAL 数据库及应用程序优化

徐 峰<sup>1</sup>, 董大勇<sup>1</sup>

(1. 中国第一重型机械集团公司技术管理部工程师, 黑龙江 富拉尔基 161042)

摘要: 论述如何针对 ORACAL 数据库及应用程序的优化。

关键词: 数据库; 应用程序; 优化

中图分类号: TP311.1380R 文献标识码: B 文章编号: 1673-3355 (2008) 01-0052-02

数据库技术对于大量复杂的信息能够进行科学高效的管理。在数据库领域中的各种应用软件大量涌现, 有 SQLServer, Sybase 等等, 但其中 ORACLE 是这个领域中最优秀也是最流行的大型数据库应用软件。我们针对在 ORACAL 中如何优化数据库及其应用程序进行了研究。

## 1 优化数据库

(1) 优化设置控制文件的个数、位置及其备份

建议 CONTROL\_FILE 初始化参数中指定多

## 3 电子档案的保管工作

目前以纸质档案为主体的档案保护工作主要针对纸质载体制成材料的耐久性、保存环境等方面来进行。实践证明纸质载体的档案在一定的保护条件下可保存达上千年, 而电子文件的物理载体寿命却远不如纸质档案, 磁性载体档案的寿命最多只有十多年, 光盘载体的寿命也不超过 100 年。另外, 纸张的损坏很容易被发现, 但电子文件载体的损坏只能由设备检验。因此, 对归档电子文件“载体生命”的保护应从其载体制成材料的物理、化学性质和保存环境入手, 着重解决保证电子文件载体的可用性的保护措施。对电子文档的载体要定期进行检测, 检测包括外观检测和逻辑检测, 外观检测主要是确认载体表面是否有物理损坏或变形, 是否清洁, 是否有霉斑的出现, 逻辑检测应在相应的电子计算机设备控制下

个文件应大于 2, 并将控制文件镜像到不同位置, 要把这些控制文件备份到安全的磁盘中。

(2) 优化设计表空间、数据文件的大小及位置

设计表空间 尽量把同一时间对磁盘的读写操作分散开, 如对一个表中数据进行更新时, 数据库将同时去读该表中的数据和该表上的索引信息, 如果把表的数据信息和索引信息都放在同一个数据文件中, 则数据库的速度将会变慢。最好是把数据信息和索引信息分别放在不同磁盘的两个数据文件中, 此时数据库对磁盘的读写操作将

进行, 采用专用或自行编制的检测软件对载体上的信息进行读取校对, 主要判定被检测载体是否需要重写或更新, 对所有检测出错的载体, 必须进行有效的修正或更新, 可随机修正, 也可用其对应的封存拷贝件进行修复。电子档案保管库房温湿度要相对稳定, 要有防火、防尘、防水、防紫外线直接照射功能, 要清洁、通风, 电子档案载体应远离强磁场和有害气体。为防万一, 电子档案应有多重备份。

## 4 结 语

在世界信息技术迅猛发展的趋势下, 档案工作正进入电子文件与电子档案管理的新阶段, 档案工作者应该审时度势, 提高驾驭电子档案的管理能力, 使电子文件在档案信息开发和利用中发挥更大的作用。

分散在两个磁盘上,速度将得到显著提高。因此在设计数据库的表空间和数据文件时,首先给表和表的索引分别创建两个表空间(存放用户数据的数据表空间和存放表索引的索引表空间)。另外,还根据该系统的数据量的大小及系统中的数据的性质不同,再考虑创建几个数据表空间或者给数据表空间添加几个数据文件。

**设计数据文件的大小** 把记录大小相当的表放在同一个表空间中,这时一个表空间的存储参数设置,可以保证表中的记录都放在一个范围内,避免了一条记录跨范围存放,可以明显提高数据库的性能。

**设计数据文件的位置** 为了避免磁盘的 I/O 操作冲突,应把数据文件创建在不同位置。

### (3) 优化设计重演日志文件

**重演日志文件的大小** 由于数据库在利用重演日志文件时是循环使用它们的,而且当 LGWR 进程在两个日志文件切换时,将自动产生一个检测点,所以重演日志文件的大小会直接影响到检测点出现的频率。而由于在数据库检测点时,对用户而言,数据库的速度会受影响,所以检测点的出现频率大,或者检测点正好出现在数据库处理数据高峰期,将会极大影响数据库的性能。因此,重演日志文件的大小设计,应考虑检测点出现的频率以及应避开数据库处理数据的高峰期。

**重演日志文件组的个数** 在 ARCHIVELOG 模式下时,适当增加重演日志文件组的个数,可以降低数据库存档日志文件的频率。

**重演日志文件的存档** 应把重演日志文件的存档之处设置在磁盘读写更快的物理设备上。这样可以减少日志文件的存档时间。

**回滚段的优化设计** 回滚段数目的设计,回滚段大小的设计 要考虑该回滚段能容纳数据库可能的最大事务的回滚项数。

**优化数据库使用的内存配置** 要调整数据库的高速缓冲区,共享池、重演日志缓冲区和进程全局区的大小平衡问题。

## 2 优化应用程序

### 2.1 采取统一的 SQL 程序

因为 ORACLE 在执行 SQL 文时,首先将当

前要执行的 SQL 文与公共区域中保存的先前执行过的 SQL 文进行比较,SQL 文相同时,就会跳过当前要执行的 SQL 文的解析处理,这样通过减少解析次数就加快 SQL 文的执行速度。

### 2.2 选取合适的 SQL 代码

在进行数据库操作时,同一结果可以用很多方法来实现。我们应当选取合适的 SQL 代码,使其更容易明了,并且运行速度更快。

### 2.3 明确列名

使用 SELECT 取得的列名一定要明确指定,并且应当使抽出的记录要尽量少。用 SORT 等的场合,因为 ORACLE 将必要的列的值放到 WORK 领域,因此减少列数也就节约了 I/O 回数。

### 2.4 注意结合索引的顺序

由于列的顺序不同而导致索引的作用效果不同。例如, (AGE, ADDRESS) 顺序的结合索引。

WHERE AGE=28 AND ADDRESS= 'BEIJING'  
& 结合索引有效

WHERE AGE=28 & 结合索引有效

WHERE ADDRESS= 'BEIJING' & 结合索引无效

### 2.5 注意 WHERE 的使用

#### (1) 索引列不要使用函数

由于索引列一旦使用了函数,索引就会变无效,因此在能用索引的情况下,尽量改变程序来利用索引。如, WHERE SUBSTR (TELNO, 2, 2) = '88' 改为 WHERE TELNO LIKE '88%' 较好。

#### (2) 索引列不要使用 NOT

由于索引列一旦使用不等于,索引就会变得无效,最后导致速度变慢。

#### (3) 不要用 NULL

如果用了 NULL,索引就会无效,变成全表检索,影响处理速度。

#### (4) 不要对索引列进行计算

如果对索引列进行计算,索引就会无效,也会导致速度变慢。

## 3 结 语

本文所提出的优化方法,能够起到提高 ORACLE 数据库性能以及提高 ORACLE 应用程序执行效率的作用。这仅是我们在实际工作中的经验总结,希望能够对广大同行起到一定帮助作用。

