

# ORACLE 数据库中提高时间响应的几种方法

刘志英 (中国科学院空间应用与研究中心 100080)

**摘要:**本文从数据库表的设计、数据录入到数据查询围绕提高时间性能作了讨论,并通过模拟数据库的运行对时间响应性能进行估价,从而调整和修改原有设计以达到较为理想的效果。

**关键词:**ORACLE 数据库, 时间效率

## 1. 引言

数据库设计需要综合考虑数据结构、存取路径、时间、空间等一系列问题以提高数据库的应用性能。对于具有实时性要求、数据量比较大的系统来说,提高时间响应就显得尤为重要,有时甚至要以空间换取时间以满足要求。例如对于一个需要定时接受大量传输数据并进行实时处理的系统,处理速度是大家非常关心的问题。ORACLE 数据库在大型数据库管理中保证数据安全性、一致性及数据共享等优越性是毋庸置疑的,而对于大量数据的存取速度却是许多建立类似系统的人们遇到的一个问题。影响时间的因素很多,它和所存取的内容和调用方式、所使用的逻辑数据结构、存储结构、选取的存储分配参数、设备的 I/O 特点、每次的 I/O 时间等有关,由于实际需要,笔者在工作过程中为提高数据存取的响应时间从下面几个方面作了许多实验比较,取得了一定效果。

## 2. 设计数据结构

存储结构格式和数据分布影响着数据存取性能,因而必须首先合理设计数据结构,数据在库中应尽量压缩存储以减少冗余及 I/O 次数。因为表中记录数越多插入和查询速度越慢,所以设计的数据结构应使得记录条数在能够方便查询的基础上尽可能少,例如若每个参数在同一时刻有两百个值,如果只按时间查询,可以把它们的原码作为一个字段用字符存储,这样大大地减少了时间开销。也可以按照所处理数据的性质将其适当压缩存储,从数据库取出后再进行压缩。

## 3. 建立数据库表

在数据库表比较多又需要经常对其进行整理、更新的情况下,用 PRO \* C 建立、管理这些表比较方便,每次作完数据备份后可以把表和数据一起删掉重新建表,这

样既可提高操作速度又能消除链接块。在建表时根据需要指定 PCTFREE、PCTUSED、STORAGE 等参数来控制数据块的使用情况,一方面提高数据插入和修改的效率,另一方面也要提高存储空间的使用效率从而整体地改善数据存取性能。对于时间要求高、空间资源丰富的系统适当考虑牺牲空间来换取时间,所以可把 PCTFREE 设置较高而 PCTUSED 较低以缓解数据在 INSERT 期间重新组织的处理开销,这时行数据跨越多个数据块的可能性减少也改进了数据检索性能。STORAGE 描述每个表和索引所占据的磁盘空间的数量,若事先估算出每个表的最大体积,在生成表时可设置存储参数使表的初始扩展能容下当前所有数据,这样所用表的数据被存在连续的磁盘空间中,这将减少有关该表的 I/O 磁盘操作时间,减少数据块的不连续性,从而使检索速度显著提高。

估算一个表的需的磁盘空间公式如下:(非聚族)

$$\text{blocks} = \frac{X \text{ rows} * (5\text{bytes} + Y \text{ cols} * (1 \text{ bytes} + Z))}{(\text{blocksize bytes} - 90\text{bytes}) * (1 - \text{PCTFREE}/100)}$$

此处:

blocks: 数据块数

X: 表中行数

Y: 表中列数

Z: 平均列长度

blocksize: DB 块大小,本系统中为 2048 bytes

上面结果乘以 blocksize(2048)即得此表所需字节总数。blocksize 的值可以从 init. ora 中查找或用 SQL \* DBA 命令查询:

```
SQLDBA>connect internal;
```

```
SQLDBA>show parameter db-blocks-size;
```

这样算出的只是表的近似体积,并不完全准确,还可以用下面的方法估算:

先建立起与将要使用的表结构完全相同的表,再装

入一定数量的数据,然后查看模拟数据已占用了多少块,用下面的 SQL 命令:

```
SQL>select count(DISTINCT(substr(ROWID,1,8)
||
substr(WOWID,15,4)))from 表名;
```

ROWID 字符串的前八个字符表示块号,后四个字符表示文件号,||是字符串联接运算,也就是计算不同数据文件中不同数据块的数目,然后估计最终需装入的记录行数,估算出实际所需空间的大小。用命令:

```
SQL>select * from taballoc;
```

也可查看表所占的空间数、索引文件和数据文件所占初始空间数。

#### 4. 存储空间安排

存储空间安排要在数据库管理员的协助下完成。DBA 根据程序开发者提出的要求定义一系列表空间来存储各种数据,根据每部分数据量估算出每个表空间中指定的数据文件和重作记录文件的大小。建立多个表空间首先是为了分隔开不同种类的信息,例如可以把表数据和索引数据分放在不同的表空间以减少 I/O 冲突,因为这样可以规划表数据和索引数据在不同的磁盘上,在查询表信息时不同的磁盘能同时工作,提高系统性能。

#### 5. 合理建立索引

在数据库表上建立适当的索引能够提高排序、检索及统计速度。一般情况下先插入数据然后建立索引,但在需要实时显示数据的情况下数据插入的同时也在查询检索,所以应该在建表的同时建立索引,虽然这样会影响插入速度。建立索引时指定其在索引空间内,并指定 NOSORT 选项。NOSORT 的作用是当索引项已是升序时时,允许 ORACLE 避免索引过程中进行排序,从而相当程度地减小重组索引需要的时间和空间开销。在一个上万条记录的表中用聚组函数统计某一参数的和要花半小时之久,如果对查询的这项建立索引,加上建索引的时间也不过 3 分钟,大大提高了统计速度。但索引的列数也不能太多,查询时经常在条件子句中使用的列要建立索引,只用于显示数据值的列就没必要建立。另外检索时引用索引项不要被函数或算术运算改变,即索引项不要引用在函数内部,否则索引不起作用。

建立不同的索引测试其运行效率从而找到最佳的索引方案。

#### 6. 确定提交频率

插入表中的数据提交之前存在每个用户的特定工作区中,对库中的数据无任何影响。在大量数据的插入中,如果提交太频繁,ORACLE 连续的执行 COMMIT 命令,比较浪费时间,提交次数太少、工作区过于拥挤、回退段不够用也会使其效率下降,而且在提交之前其他用户无法检索到新改变的数据,所以应逻辑合理地使用 COMMIT 语句及其 RELEASE 选项,从而避免阻碍其他用户对数据库的存取。提交频率的确定没有明确的理论准则,可以根据多次实验通过测定不同频率下的运行时间确定。本实验的结果是时间效率和提交频率关系曲线类似高斯曲线,从而选取最佳频率。

#### 7. 数据量控制

ORACLE 数据库原则上对数据量没有限制,能容纳数据的多少只取决于硬盘,可是即使硬盘容量很大,也不能在硬盘中存放过多的数据。存取一个装满磁盘中的数据比检索一个自由空间较多的磁盘上的数据所需 I/O 次数明显要多。一个表中数据行数过多插入和检索该表中数据速度都会明显降低。考虑实际需要,将数据控制在较小的规模,过一段时间或者数据达到一定量就作一次转储,完成转储后把所有数据库表 DROP 掉重建,这样一是可以提高时间和空间利用率,二是消除数据碎片。

#### 8. 小结

ORACLE 的数据管理机制决定其在速度上有所限制,但如果各种结构设计合理,甚至在其他性能上作出一些让步的话,还是可以达到某些要求的。

#### 参考文献

- [1] Administrator the ORACLE Database I(ORACLE7), Copyright Oracle Corporation, 1993, Printed in the USA.
- [2] 陶辅周等,实用 ORACLE 系统开发和设计详解,电子工业出版社,1995。
- [3] 沈佩娟等,ORACLE 数据库管理及应用开发,清华大学出版社,1995。