

文章编号: 1005-8451 (2005) 12-0046-02

## Oracle 数据库碎片整理

田建军

(广东三茂铁路股份有限公司 电算中心, 广东 510060)

**摘要:** 提高数据库性能是数据库管理员一直追求的目标, 减少数据库碎片是提高系统运行效率和工作性能的重要方法之一。介绍数据库碎片的产生、计算方法及整理方法。

**关键词:** 数据库; 表空间; 空闲空间; 整理

中图分类号: TP39

文献标识码: B

Oracle 作为一种大型数据库, 广泛应用于铁路各应用系统中。对于系统管理员来讲, 如何保证网络稳定运行, 如何提高数据库性能, 使其更加安全高效, 就显得尤为重要。作为影响数据库性能的一大因素—数据库碎片, 应当引起 dba 的足够重视, 及时发现并整理碎片是 dba 一项基本维护内容。

### 1 碎片是如何产生的

当生成一个数据库时, 它的逻辑结构包括表空间 (tablespace) 逻辑段 (segment) / 区 (EXTENTS) / 数据块 (DATA BLOCKS), 图 1 表明了数据中各逻辑单位之间的关系。数据库在逻辑上是由多个表空间构成的, 表空间在物理上包含一个或多个数据文件, 而数据文件是数据块的整数倍。表空间中存储的对象叫段, 而段是由区组成的, 区是磁盘空间分配的最小单位, 段的增大 (比如基表中的数据增加) 是通过增加区的个数来实现。在分配磁盘空间时, 至少每次分配一个区。每个区的大小必须是数据块的整数倍, 区的大小可以不相同, 但都是数据块的整数倍, 而数据块数据库中最小的 I/O 单位, 同时是内存数据缓冲区的单位及数据文件存储空间单位。

表空间、段、区、空闲区的逻辑关系如图 1 所示。

当表空间中生成一个段时, 将从表空间有效自由空间中为这个段的初始范围分配空间。在这些初始范围充满数据时, 段会请求增加另一个范围 (一个或多个区)。这样的扩展过程会一直继续下去, 直到达到最大的范围值, 或者在表空间中已经没有自由空间用于下一个范围。最理想的状态就是一个段的数据可被存在单一的一个范围中。这样, 所有的

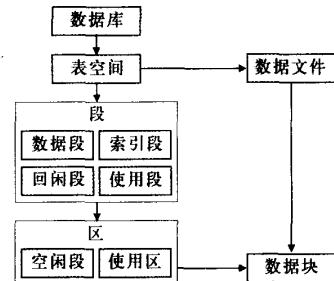


图 1 数据库逻辑结构

数据存储时靠近段内其它数据, 并且寻找数据可少用一些指针。但是一个段包含多个范围的情况是大量存在的, 没有任何措施可以保证这些范围是相邻存储的。当要满足一个空间要求时, 数据库不再合并相邻的自由范围 (除非别无选择), 而是寻找表空间中最大的自由范围来使用。这样将逐渐形成越来越多离散的、分隔的、较小的自由空间, 即碎片。

### 2 碎片对系统的影响

(1) 导致系统性能减弱: 如上所述, 当要满足一个空间要求时, 数据库将首先查找当前最大的自由空间, 而“最大”自由空间逐渐变小, 要找到一个足够大的自由空间已变得越来越困难, 从而导致表空间中的速度障碍, 使数据库的空间分配愈发远离理想状态。

(2) 浪费大量的表空间: 尽管有一部分自由空间 (如表空间的 pctincrease 为非 0) 将会被 smon (系统监控) 后台进程周期性地合并, 但始终有一部分自由空间无法得以自动合并, 浪费了大量的表空间。

### 3 自由表空间的碎片计算

由于自由空间碎片是由几部分组成, 如自由空

间数量、最大范围尺寸等，我们可用 `fsfi--free space fragmentation index` (自由空间碎片索引) 值来直观体现： $fsfi=100*\sqrt{\max(extent)/\sum(extents)}*1/\sqrt{\sqrt{\sum(count(extents))}}$

可以看出，`fsfi` 的最大可能值为 100 (一个理想的单文件表空间)。随着范围的增加，`fsfi` 值缓慢下降，而随着最大范围尺寸的减少，`fsfi` 值会迅速下降。

下面的脚本可以用来计算 `fsfi` 值：

```
rem  fsfi value compute
rem  fsfi.sql
column fsfi format 999,99
select tablespace_name,sqrt(max(blocks)/sum(blocks))*
(100/sqrt(sqrt(count(blocks)))) fsfi
from dba_free_space
group by tablespace_name order by 1;
spool fsfi.rep;
/
spool off;
```

--- 比如，在某数据库运行脚本 `fsfi.sql`，得到以下 `fsfi` 值：

| tablespace_name | fsfi   |
|-----------------|--------|
| rbs             | 74.06  |
| system          | 100.00 |
| temp            | 22.82  |
| tools           | 75.79  |
| users           | 100.00 |
| user_tools      | 100.00 |
| ydcx_data       | 47.34  |
| ydcx_idx        | 57.19  |
| ydfj_data       | 33.80  |
| ydfj_idx        | 75.55  |

统计出了数据库 `fsfi` 值，可以把它作为一个可比参数。在一个有着足够有效自由空间，且 `fsfi` 值超过 30 的表空间中，很少会遇见有效自由空间问题。当一个空间将要接近可比参数时，需要做碎片整理。

## 4 表空间的碎片整理

空间的 `pctincrease` 值为非 0

可以将表空间的缺省存储参数 `pctincrease` 改为非 0。一般将其设为 1，如：

```
alter tablespace temp
default storage(pctincrease 1);
```

这样 `smon` 便会将空闲区自动合并。

(2) 也可以手工合并表空间中相邻的空闲区：  
`alter tablespace temp coalesce。`

## 5 段的碎片整理

段由区组成，有必要对段的碎片进行整理。要查看段的有关信息，可查看数据字典 `dba_segments`，区的信息可查看数据字典 `dba_extents`。如果段的碎片过多，将其数据压缩到一个范围的最简单方法便是用正确的存储参数将这个段重建，然后将旧表中的数据插入到新表，同时删除旧表。这个过程可以用 `import/export` (输入/输出) 工具来完成。

`export` () 命令有一个 (压缩) 标志，这个标志在读表时会引发 `export` 确定该表所分配的物理空间量，它会向输出转储文件写入一个新的初始化存储参数等于全部所分配空间。若这个表关闭，则使用 `import` () 工具重新生成。这样，它的数据会放入一个新的、较大的初始段中。例如：

```
exp user/password file=exp.dmp compress=y
grants=y indexes=y
tables=(table1,table2);
```

若输出成功，则从库中删除已输出的表，然后从输出转储文件中输入表：

```
imp user/password file=exp.dmp commit=y
buffer=64000 full=y
```

这种方法可用于整个数据库。

## 6 结束语

以上简单分析 Oracle 数据库碎片的产生、计算方法及整理。在 Oracle9i 中，系统区别使用本地化管理表空间，所以不再存在磁盘碎片问题。但铁路很多应用系统如车站系统、调度系统等都使用的是 Oracle 7 和 Oracle8，在磁盘存有大量碎片需要数据库管理员解决问题。数据库的性能优化是一项技术含量高，同时又需要有足够耐心、认真细致地工作。

参考文献：

- [1] 腾永昌. Oracle9i 数据库管理员使用大全 [M]. 北京：清华大学出版社，2003.
- [2] Jonathan Gennick. Oracle 8i DBA 宝典 [M]. 北京：电子工业出版社，2004.