

# ORACLE 数据库 空间管理实践

## Management and Practice of Oracle Database Space



徐正雄 王玲 (成都军区昆明总医院信息科 650032)

**摘要:** 本文对 Oracle 数据库空间管理中表空间难以回收的问题进行研究, 提出了利用 Export/Import 工具有效回收数据库表空间的方法和措施。

**关键词:** Oracle 数据库 表空间管理

因为大型 Oracle 数据库系统具有海量数据存储及变化的特性, 随着数据库的使用和数据量的变化, 势必会出现许多数据碎片, 数据空间无限增长, 硬盘存储空间缩小, 整个数据库性能降低等问题。所以 ORACLE 数据库空间的有效管理对整个数据库系统的正常运行以及性能优化具有十分重要的作用。笔者在从事 ORACLE 数据库管理工作期间, 总结了一些 ORACLE 数据库空间管理经验, 供同行参考。

### 1 有关概念

数据库的空间在逻辑上分为多个表空间, 每个表空间则由系统中的一个或多个物理数据文件构成; Oracle 存储数据的基本单位是块, 其大小在建库时由 DB\_BLOCK\_SIZE 参数确定, 一个或多个连续的块构成一个区片 (EXTENT), 它作为数据对象存储的基本单位来使用。在 Oracle 中, 每个基本数据对象使用的空间称为段 (Segment), 段存放在唯一的表空间上, 每个段实际上是一系列区片 (更为准确的表达称数据块) 的集合。每个简单数据对象对应一个段; 对于分区对象如分区表、索引, 则每个 (子) 分区对应一个段, 由各个 (子) 分区共同构成一个完整的数据对象。

### 2 表空间碎片的产生

一个段由范围组成, 而范围是由相邻的 Oracle 块组成。当 Oracle

向段中写入数据时, 它会更新段的 High-water (高位) 标记。High-water 标记指明了段中存储数据块的最高值。如果数据表中插入 1000,000 块数据值, 然后使用 delete 命令删除这些记录, 那么数据表中就没有记录行了。但 High-water 标记还设置在 1000,000 处。下次对这个数据表执行全表扫描时, oracle 将扫描 1000,000 个空的数据块。High-water 标记之上的段空间可以使用 "alter table spaces deallocate unused keep<需要保留自由空间量>" 命令回收, 而 High-water 标记之下的段空间用删除记录方式是不会回收表中空间的。如果表中记录被频繁的 DELETE 或 INSERT, 尽管表中的记录总量不变, 但表空间的使用量仍会不断增加。一旦存在的范围不能再存储新的数据, Oracle 段就会去获取新的范围。这样的扩展过程会一直继续, 直到表空间中的数据文件不能提供更多的自由空间, 或者范围数量已达到最大值。Oracle 数据库管理员都有这样的经历, 对一个大数据量表的记录进行删除之后, 表空间占用 (USED) 率未见缩小, 运行效率未见提高。因此一个碎片太多的数据段, 不仅会影响运行效率, 也会影响表空间中的空间管理问题。

### 3 表空间碎片的消除

表空间碎片的消除, 可以使用 Export/Import (输出/输入) 工具来很好地完成。当我们对历史数据进行清除 (Delete) 之后, 可以使用

Export (输出) 工具对数据执行输出。但在Export参数中没有 "Tablespace" 这个参数, 使用帐户(Owner)输出可以产生所希望的结果。现在假设有三个帐户 (Inpbill, Pharmacy, Outpbill) 的情况, 我们在System下调用SQL语句, 对DBA\_TABLES和DBA\_INDEXES数据字典视图进行查询:

```
select tablespace_name,owner,count(*) || 'tables' objects
from DBA_TABLES
group by tablespace_name,owner
union
select tablespace_name,owner,count(*) || 'indexes' objects
from DBA_INDEXES
group by
tablespace_name,owner;
```

查询结果如下:

TABLESPACE_NAME	OWNER	OBJECTS
TSP_INPBILL	INPBILL	19tables
TSP_INPBILL	INPBILL	24indexes
TSP_PHARMACY	Pharmacy	79tables
TSP_PHARMACY	Pharmacy	83indexes
TSP_PHARMACY	Outpbill	2tables
TSP_OUTPBILL	Outpbill	11tables
TSP_OUTPBILL	Outpbill	15indexes

以上输出样本显示, TSP\_INPBILL表空间仅包含一个帐户对象 (INPBILL帐户), 而TSP\_PHARMACY表空间包含了两个帐户的表和索引。由于TSP\_INPBILL表空间仅含INPBILL帐户, 所以输出(Export)INPBILL帐户就输出了TSP\_INPBILL表空间的所有对象。而对于TSP\_PHARMACY表空间, 可以通过查询数据字典决定Outpbill帐户所包含的TSP\_PHARMACY表空间中的表名, 然后通过Export命令的 "Tables" 参数输出这些表, 通过帐户PHARMACY输出TSP\_PHARMACY表空间其余的对象。

现在以TSP\_INPBILL表空间为例进行说明:

```
exp system/manager file=c:\inpbill.dmp log=c:\export.log grants=y
owner=INPBILL buffer=1000000 compress=N
```

从上述举例可以观察到我们使用compress=N这个参数设置, 这与很多数据库管理员经常使用的compress=Y这一ORACLE缺省参数设置不同, 这是本文中想重点强调的地方。Compress参数是指导出数据时是否将表数据压缩到一个期间的标志, 单纯地从字面上看也是“压缩”的意思, 并且ORACLE的缺省设置也为Y, 所以很多数据库管理员忽略了这点, 将数据Export (compress=Y) 后, 在Import时却发现碎片数据段压缩了, 但多余的数据空间却没有回收。这是因为compress=Y压缩

的空间是定位, 而不是使用。在三个100MB位置的300MB表, Export (compress=Y) 后, 将压缩到300MB位置。若表空间有多个数据文件, compress=Y改变STORAGE字句使初始空间尺寸大于任何一个数据文件。简而言之, 如果TSP\_INPBILL表空间USED为300MB, Export (compress=Y) 后, 执行Import, USED仍然为300MB, 不会回收多余空间。所以对于大数据量的表空间, 我们建议在Export过程中使用compress=N参数, 并在创建表空间时AUTOEXTEND参数置为ON。按上述方法, 在实践中取得了回收多余空间, 合并数据碎片的很好效果。

当Export执行完毕后, 表空间就可以删除并重建, 并可输入完整的输出文件。

```
SVRMGR>connect internal
```

```
SVRMGR>drop tablespace tsp_inpbill;
```

```
SVRMGR> create tablespace tsp_inpbill
```

```
datafile 'd:\oracle\oradata\apinpbill.dbf'
```

```
size 50M
```

```
reuse
```

```
autoextend on
```

```
next 20M
```

```
maxsize unlimited;
```

```
imp system/manager file=c:\inpbill.dmp full=y log=c:\imp.log
ignore=y commit=y buffer=10000000
```

执行完毕后, 就可以观察到TSP\_INPBILL表空间被有效地回收, 数据库性能得到有效的提高。经过执行以上的数据管理方法, 解决了困扰我们的Oracle数据库表空间难以回收的问题。目前, 我院的数据库运行顺畅, 取得了较好的管理效果。

以上整个过程在WIN2000, ORACLE8.1.6环境中测试通过。

## 参考文献

- 1 Oracle8数据库管理员手册, (美)Kevin Loney 著, 李晓军等译, 机械工业出版社, 1998。