**通过 dbms\_space.space\_usage 查看objects 的空间使用情况**

[上一篇](http://space.itpub.net/batch.common.php?action=viewspace&op=up&itemid=674472&uid=35489) / [下一篇](http://space.itpub.net/batch.common.php?action=viewspace&op=next&itemid=674472&uid=35489)  2010-09-20 13:59:06 / 个人分类：[Oracle10g RAC](http://space.itpub.net/35489/spacelist-blog-itemtypeid-16040)

[查看( 174 )](http://space.itpub.net/35489/viewspace-674472#xspace-tracks) / [评论( 1 )](http://space.itpub.net/35489/viewspace-674472#xspace-itemreply) / [评分( 0 / 0 )](http://space.itpub.net/35489/viewspace-674472#xspace-itemform)

SQL> show serveroutput  
serveroutput OFF  
SQL>  
SQL> set serveroutput on;  
SQL> show serveroutput  
serveroutput ON SIZE UNLIMITED FORMAT WORD\_WRAPPED  
SQL>   
declare  
     l\_fs1\_bytes number;  
    l\_fs2\_bytes number;  
    l\_fs3\_bytes number;  
    l\_fs4\_bytes number;  
    l\_fs1\_blocks number;  
    l\_fs2\_blocks number;  
    l\_fs3\_blocks number;  
    l\_fs4\_blocks number;  
    l\_full\_bytes number;  
    l\_full\_blocks number;  
    l\_unformatted\_bytes number;  
    l\_unformatted\_blocks number;  
  begin  
      dbms\_space.space\_usage('DFMS',  
         'IDX1\_WIP\_D\_WO\_DETAIL\_COMID',  
         'INDEX',  
         fs1\_bytes=> l\_fs1\_bytes,  
         fs1\_blocks         => l\_fs1\_blocks,  
         fs2\_bytes          => l\_fs2\_bytes,  
         fs2\_blocks         => l\_fs2\_blocks,  
         fs3\_bytes          => l\_fs3\_bytes,  
         fs3\_blocks         => l\_fs3\_blocks,  
         fs4\_bytes          => l\_fs4\_bytes,  
         fs4\_blocks         => l\_fs4\_blocks,  
         full\_bytes         => l\_full\_bytes,  
         full\_blocks        => l\_full\_blocks,  
         unformatted\_blocks => l\_unformatted\_blocks,  
         unformatted\_bytes  => l\_unformatted\_bytes   );  
      dbms\_output.put\_line(' FS1 Blocks = '||l\_fs1\_blocks||' and Bytes = '||l\_fs1\_bytes);  
      dbms\_output.put\_line(' FS2 Blocks = '||l\_fs2\_blocks||' and Bytes = '||l\_fs2\_bytes);  
      dbms\_output.put\_line(' FS3 Blocks = '||l\_fs3\_blocks||' and Bytes = '||l\_fs3\_bytes);  
      dbms\_output.put\_line(' FS4 Blocks = '||l\_fs4\_blocks||' and Bytes = '||l\_fs4\_bytes);  
      dbms\_output.put\_line(' Full Blocks = '||l\_full\_blocks||' and Bytes = '||l\_full\_bytes);  
      dbms\_output.put\_line(' Unformatted Blocks = '||l\_unformatted\_blocks||' and Bytes =

'||l\_unformatted\_bytes);  
   end;  
   /

FS1 Blocks = 0 and Bytes = 0  
FS2 Blocks = 1548 and Bytes = 25362432  
FS3 Blocks = 0 and Bytes = 0  
FS4 Blocks = 0 and Bytes = 0  
Full Blocks = 3405 and Bytes = 55787520  
Unformatted Blocks = 3405 and Bytes = 18633981952

PL/SQL procedure successfully completed.

SQL>

注意执行权限 。

-------------------------------------------------

顺便介绍一下  analyze  index    xxxx  validate structure ;  的使用。

SQL> analyze  index   dfms.IDX1\_WIP\_D\_WO\_DETAIL\_COMID  validate structure ;

SQL> select name, del\_lf\_rows, lf\_rows, round((del\_lf\_rows/(lf\_rows+0.0000000001))\*100)  
  2   "Frag Percent"  from index\_stats   where   name = 'IDX1\_WIP\_D\_WO\_DETAIL\_COMID'  ;     
NAME                                                    DEL\_LF\_ROWS        LF\_ROWS        Frag Percent  
------------------------------ ----------- ---------- ------------  
IDX1\_WIP\_D\_WO\_DETAIL\_COMID         1732765    1817931           95

[导入论坛](http://www.itpub.net/post.php?action=import&itemid=674472) [引用链接](javascript:;) [收藏](javascript:;) [分享给好友](javascript:;) [推荐到圈子](javascript:;) [管理](http://space.itpub.net/batch.manage.php?itemid=674472) [举报](javascript:;)

**TAG:**

[引用删除](javascript:;)[tolywang](http://space.itpub.net/35489/)   /   2010-09-20 14:44:47

备注：  Oracle 引入的BMB块(bitmap blocks), 数据块的使用状况分为 6 个等级，分别是unformated ,   可用空间小于块大小的25% (即0~25%) ，25%~50%， 50%~75% ， 75%~100% ， Full (数据块没有可用空间)       
这个procedure中的参数代表如下(可以在dbms\_space.space\_usage中找到注释 )：       
  
  --   unformatted\_blocks  
  --      total number of blocks that are unformatted   
  --   fs1\_blocks  
  --      number of blocks that have atleast 0 to 25% free space.  
  --   fs2\_blocks  
  --      number of blocks that have atleast 25% to 50% free space.  
  --   fs3\_blocks  
  --      number of blocks that have atleast 50% to 75% free space.  
  --   fs4\_blocks  
  --      number of blocks that have atleast 75% to 100% free space.  
  --   full\_blocks  
  --      total number of blocks that are full in the segment

**DBMS\_SPACE包的使用（1）**

[上一篇](http://space.itpub.net/batch.common.php?action=viewspace&op=up&itemid=592035&uid=22235) / [下一篇](http://space.itpub.net/batch.common.php?action=viewspace&op=next&itemid=592035&uid=22235)  2009-04-27 23:57:45 / 个人分类：[Oracle随记](http://space.itpub.net/22235/spacelist-blog-itemtypeid-72118)

[查看( 345 )](http://space.itpub.net/22235/viewspace-592035#xspace-tracks) / [评论( 0 )](http://space.itpub.net/22235/viewspace-592035#xspace-itemreply) / [评分( 0 / 0 )](http://space.itpub.net/22235/viewspace-592035#xspace-itemform)

最近有朋友问到了DBMS\_SPACE包的使用，也看了一下，大部分是关于dbms\_space.space\_usage的使用，space\_usage这个过程的例子已经很多了，我也就不再多说了，除了这个过程外，另外还有两个过程也有着特殊的用处，但使用的人不多，我们也来看看这两个过程有什么用。

这两个过程为：CREATE\_INDEX\_COST和CREATE\_TABLE\_COST，分别用户评估创建索引和创建表的存储开销（空间占用情况）。

CREATE\_INDEX\_COST的语法如下：

DBMS\_SPACE.CREATE\_INDEX\_COST (

  ddl            IN   VARCHAR2,

  used\_bytes     OUT  NUMBER,

  alloc\_bytes    OUT  NUMBER,

  plan\_table     IN   VARCHAR2 DEFAULT NULL);

下面是相关的**[测试](javascript:;" \t "_self)**[**代码**](javascript:;)：

1、准备相关表和数据

SQL> set serveroutput on

SQL> create table t(c char(100),d varchar2(200));

表已创建。

SQL> begin

 2   for i in 1..5000 loop

 3    insert into t values(i,i);

 4   end loop;

 5   commit;

 6 end;

 7 /

PL/SQL过程已成功完成。

2、分析表，注意：没有统计信息，CREATE\_INDEX\_COST将无法计算索引的存储开销

SQL> analyze table t compute statistics;

表已分析。

SQL> declare

 2   v1 number;

 3   v2 number;

 4 begin

 5   DBMS\_SPACE.CREATE\_INDEX\_COST('create index i on t(c)',v1,v2);

 6   dbms\_output.put\_line(v1/1024||' '||v2/1024);

 7 end;

 8 /

488.28125 640   --计算出的索引将占用488K字节空间，为该索引需要分配640k存储空间

PL/SQL过程已成功完成。

3、创建实际索引，确定索引存储空间是否与计算的结果相符

SQL> create index i on t(c);

索引已创建。

SQL> select count(\*) from user\_extents where segment\_name='I';

 COUNT(\*)

----------

       11

已选择1行。 --11个64k的区，比计算出的大1个区

4、再次装载数据

SQL> begin

 2   for i in 1..5000 loop

 3    insert into t values(i,i);

 4   end loop;

 5   commit;

 6 end;

 7 /

PL/SQL过程已成功完成。

SQL> declare

 2   v1 number;

 3   v2 number;

 4 begin

 5   DBMS\_SPACE.CREATE\_INDEX\_COST('create index i on t(c)',v1,v2);

 6   dbms\_output.put\_line(v1/1024||' '||v2/1024);

 7 end;

 8 /

488.28125 640          --没有分析之前，获得得仍然是根据以前分析结果计算的值

PL/SQL过程已成功完成。

SQL> analyze table t compute statistics;

表已分析。

SQL> declare

 2   v1 number;

 3   v2 number;

 4 begin

 5   DBMS\_SPACE.CREATE\_INDEX\_COST('create index i on t(c)',v1,v2);

 6   dbms\_output.put\_line(v1/1024||' '||v2/1024);

 7 end;

 8 /

976.5625 2048                 --分析之后，得到新的结果

PL/SQL过程已成功完成。

5、再次验证，16个64k的区和1个1024k的区，2048k，与估计值相同

SQL> select count(\*) from user\_extents where segment\_name='I';

 COUNT(\*)

----------

       17

-------------------------------------------------------------

6、换了一个字段进行测试

SQL> declare

 2   v1 number;

 3   v2 number;

 4 begin

 5   DBMS\_SPACE.CREATE\_INDEX\_COST('create index i on t(d)',v1,v2);

 6   dbms\_output.put\_line(v1/1024||' '||v2/1024);

 7 end;

 8 /

39.0625 192       --计算出的索引将占用39K字节空间，为该索引需要分配192k存储空间

PL/SQL过程已成功完成。

7、创建索引，新建的索引比估算的值大1个区

SQL> create index i on t(d);

索引已创建。

SQL> select count(\*) from user\_extents where segment\_name='I';

 COUNT(\*)

----------

        4

SQL> drop index i;

8、再次装载数据并分析表

SQL> begin

 2   for i in 1..10000 loop

 3    insert into t values(i,i);

 4   end loop;

 5   commit;

 6 end;

 7 /

PL/SQL过程已成功完成。

SQL> analyze table t compute statistics;

表已分析。

9、重新计算，得到新的估算值

SQL> declare

 2   v1 number;

 3   v2 number;

 4 begin

 5   DBMS\_SPACE.CREATE\_INDEX\_COST('create index i on t(d)',v1,v2);

 6   dbms\_output.put\_line(v1/1024||' '||v2/1024);

 7 end;

 8 /

78.125 320

PL/SQL过程已成功完成。

10、创建索引，新建的索引比估计的大2个区

SQL> create index i on t(d);

索引已创建。

SQL> select count(\*) from user\_extents where segment\_name='I';

 COUNT(\*)

----------

        7

11、顺便测试shink space的效果

SQL> select count(\*) from t;

 COUNT(\*)

----------

    20000

SQL> delete t where rownum<=15000;

已删除15000行。

SQL> commit;

提交完成。

SQL> alter table t enable row movement;

表已更改。

12、在删掉15000行数据后，没有整理空间之前进行统计信息收集

SQL> analyze table t compute statistics;

表已分析。

SQL> declare

 2   v1 number;

 3   v2 number;

 4 begin

 5   DBMS\_SPACE.CREATE\_INDEX\_COST('create index i on t(d)',v1,v2);

 6   dbms\_output.put\_line(v1/1024||' '||v2/1024);

 7 end;

 8 /

24.4140625 128         --基于新收集的统计信息计算，估算的索引需要分配128k存储空间

PL/SQL过程已成功完成。

13、收缩表，释放占用的存储空间

SQL> alter table t shrink space;

表已更改。

SQL> analyze table t compute statistics;

表已分析。

SQL> declare

 2   v1 number;

 3   v2 number;

 4 begin

 5   DBMS\_SPACE.CREATE\_INDEX\_COST('create index i on t(d)',v1,v2);

 6   dbms\_output.put\_line(v1/1024||' '||v2/1024);

 7 end;

 8 /

24.4140625 128  --收缩后重新收集统计信息，与原统计信息一样，因此计算出的大小一样

PL/SQL过程已成功完成。

SQL> select count(\*) from user\_extents where segment\_name='I';

 COUNT(\*)

----------

        7

--现有索引并没有收缩，仅仅是表空间进行了收缩，因此现有索引仍保持原大小

14、重建索引，对比新的索引大小与计算出的索引大小一样大

SQL> alter index i rebuild;

索引已更改。

SQL> select count(\*) from user\_extents where segment\_name='I';

 COUNT(\*)

----------

        2

--重建索引后新的索引占用空间与计算出的空间一样大