How to short tablespace size

Ciolac Anatol

04 April 2006

Introduction

The organization of space managment in Oracle are developed so, that it is very easy to add space to the database, but shorting them are more harder. For adding space to the some tablespace it is enough to execute simple SQL command :

ALTER TABLESPACE *table_space_name* ADD DATAFILE *'datafile_name'* [...] ;

Also for clearing free space there are necessary much more operations and calculations.

In this article I will describe how to minimize size of database after deleting a part of information from them. This situation often occurs when was created a copy of production database for developing tasks, but this database have great size and need to be shorted. More precisely, this article resolves a problem, when in database there are some free space, and is necessary to short database and free for Operating System free space from database. Things described in this article are not applicable in the database with **Automatic Storage Management**.

1 Organization of space in the Oracle database

The most elementary unit of disk space is the **block** which are sets generally at the time of creation of database. Space occupied by table or index on the disk are named **segment**. If table or index are partitioned, then every partition have his own segment. When segment need to grow, he grows by **extents** which consist from a number of blocks. Size of extent depends from many parameter such as NEXT_EXTENT parameter of table or index, type of managing of tablespace(**LMT** or **DMT**), etc. Block which are part of one extent are situated in database one after one. Every segment can be placed only at one tablespace. For more details about block, segments and extents see *Oracle Database Concepts* manual. We will short database by shorting of tablespaces. At the figure 1 are scheme of typical tablespace.

From the figure we see that every tablespace consist from datafiles, which con-

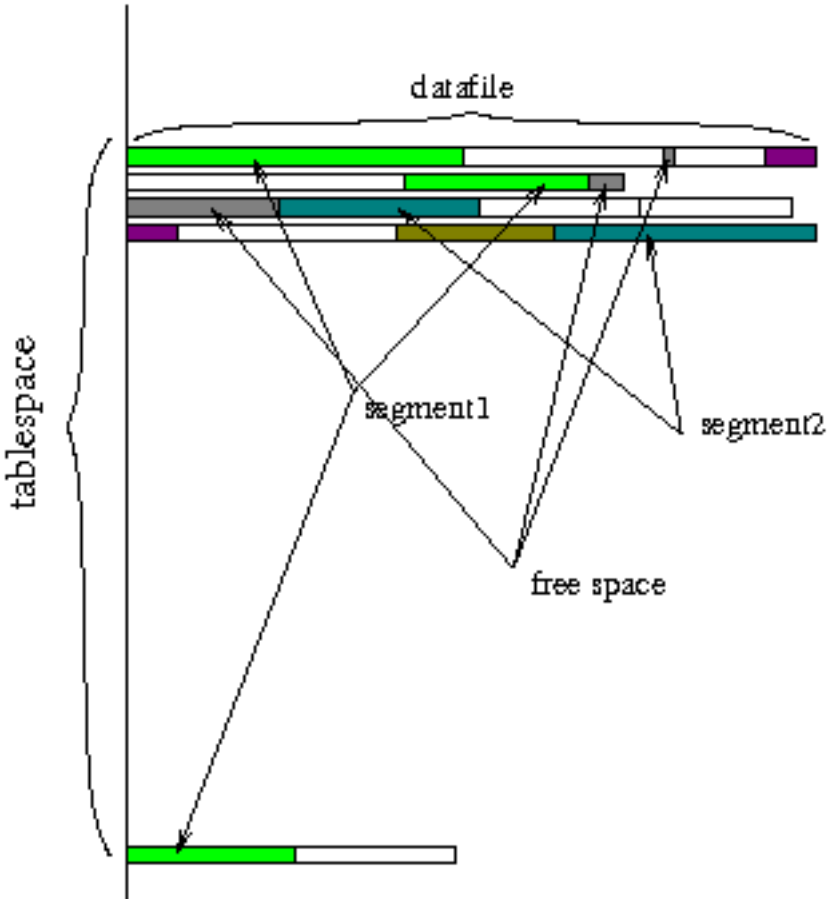

Figure 1: Scheme of space allocation in Oracle

tains segments. Segments consist from extents – every segment can have extents in any datafile of tablespace. Shorting of tablespace means shorting of datafiles. But for shorting of any datafile, it is necessary that free space, which have datafile will be placed at the end of the file(at the right side of datafile on figure 1).

2 Procedure of shorting

If we want to free space from the end of particular datafile, we must to move all segment, containing the last extent at the end of this datafile, to the other tablespace[1] The ideal case when is possible to create new tablespace which can contain all segments from minimizing tablespace. In this case is necessary next simple steps:

1. move all segments from minimizing tablespace to the new tablespace.
2. short datafiles of minimizing tablespace to the summary size of all moved segments.
3. move all segments from new tablespace to the minimized tablespace.
4. drop additional created tablespace.

But not every time is possible to create so big tablespace which can include all segments from minimizing tablespace. In this case we must create additional tablespace with how much possible big size. In the rest of text I will notice by TBLSPS1 – the tablespace which must be minimized and TBLSPS2 – tablespace which was created temporarily for helping to minimize. The algorithm of minimizing size of TBLSPS1 will be the next:

1. Select the segments, which is more convent for us for moving. this is the segment which have summary size smaller or equal to size of TBLSPS2 and moving of them get to us biggest summary free space at the end of datafiles of TBLSPS1.

[1] At the level of user, Oracle can operate only by segments, not by extents or blocks - so we cannot move only some extents.

2. Move selected segments to the tablespace TBLSPS2. This step is performing by executing the

   ALTER {TABLE|INDEX|PARTITION} *segment_ name* MOVE TABLESPACE TBLSPS2;

   for every selected segment.

3. Short datafiles of TBLSPS1. The next select generates SQL statements which must be executed on database for shorting datafiles until the last occupied block.

   ```
   SELECT 'ALTER DATABASE DATAFILE "'||file_name||"' RESIZE '||(mbid+1)||'M;'
   FROM(
   SELECT file_name, trunc(mbid/(1024*1024)) mbid, TRUNC (fl.BYTES / (1042 * 1024)),
   TRUNC ((fl.BYTES - mbid) / (1042 * 1024)) free
   FROM dba_data_files fl,
   (SELECT bl.file_id, bl.max_block_id * 8192 + dbex.BYTES mbid
   FROM dba_extents dbex,
   (SELECT ex.file_id, MAX (ex.block_id) max_block_id
   FROM dba_extents ex
   where tablespace_name='TBLSPS1'
   GROUP BY ex.file_id) bl
   WHERE bl.file_id = dbex.file_id AND bl.max_block_id = dbex.block_id) max_set
   WHERE fl.file_id = max_set.file_id
   and TRUNC ((fl.BYTES - mbid) / (1042 * 1024)) > 3[2]
   ORDER BY free DESC);
   ```

   This select generates statements of this kind:

   ```
   ALTER DATABASE DATAFILE filename1 RESIZE 20M;
   ALTER DATABASE DATAFILE filename2 RESIZE 213M;
   .........
   .........
   ```

[2] file must have at last 3MB free at the end

.....
ALTER DATABASE DATAFILE filenameN RESIZE 340M;

Execute every one of them in the database by SQL*Plus. Be sure that after this procedures in the tablespace TBLSPS1 is enough free space for segments from TBLSPS2. If it is not enough, then is necessary to add space and in this case we are minized tablespace how much possible and solved our problem. If tablespace TBLSPS1 is DMT(**Dictionary Managed Tablespace**), then there are must be space for segments with any size of extents.

4. Move segments from TBLSPS2 to TBLSPS1. This step is performing by executing the

   ALTER {TABLE|INDEX|PARTITION} *segment_ name* MOVE TABLESPACE TBLSPS1;

   for every segment in TBLSPS2. For LMT(**Locally Managed Tablespace**) tablespace there are no difference between the order of moving segments. In the case of DMT at fist necessary to move segments with larger NEXT_EXTENT

Repeating this steps minimize our tablespace TBLSPS1. As we see, the most problematical point is to select segments for move. Optimizing this selection give us economy in time and database resources.

3 Algorithm of selection based on the last extent in datafile

The easiest method to select segments - is:

1. Calculate which segment have biggest summary space in extents placed at the end of datafiles.
2. Move them to TBLSPS2
3. Repeat 1. – 2. until there are exist enough space in TBLSPS2. This points will implement point 1. and 2. of general algorithm(selecting segments and moving them in TBLSPS2).
4. Short datafiles how much is possible.

5. Move segments in TBLSPS1.

Of course, this method have many disadvantages – in the case of segments with big differences in sizes method become very ineffective, in DMT also effectivity is small whem extent sizes are very different, etc. Also this is the simplest way to resolve our problem. So I write detailed description of them:

1. The next select give us the segments name which is much more convenient for us to move.

```
SELECT segment_name,count(*)
FROM(
SELECT segment_name
FROM dba_extents ex,
(
SELECT file_id,max(block_id) max_block_id
FROM dba_extents
WHERE tablespace_name='TBLSPS1'
GROUP BY file_id
) max_ex
WHERE ex.file_id=max_ex.file_id
AND ex.block_id=max_ex.max_block_id
)
GROUP BY segment_name
ORDER BY count(*) desc;
```

2. Move the first selected segment to the TBLSPS2 by ALTER statement from p. 2 at the page 4
3. Repeat 1. – 2. until there are free space in TBLSPS2. When we see the segment which is bigger than free space in TBLSPS2, then we stop repeating.
4. Short datafiles. This point are performing some as p. 3 at the page 4.
5. Move segments back to TBLSPS1 – see p. 4 at the page 5.

4 The most effective algorithm

This part is now in the development state. I will write a PL/SQL program which resolve problem in most effective manner.