Oracle Database Tips – Index Series

Determining when an Index Should be Rebuilt?

(*version: Oracle Database 10g and later+*)

This article tries to decribe when an index should be rebuilt and the implications of rebuilding indexes.

Possibly there are many discussion about whether rebuilding indexes is useful or not. Generally, the need to rebuild **B-Tree** indexes is very rare, basically due to a **B-Tree** index is totally *Self-Managed* and *Self-Balanced.*

The most common reasons and justifications provided for rebuilding an index are:

- *Index becomes fragmented*
- *Index grows continuously and deleted space is not reused*
- *Index clustering factor becomes out of sync*

Important notes:

- Actually, most indexes remain both balanced and fragmented-free, because the free leaf entried will be reused.
- The Clustering Factor reflects how sorted the table data is with respect to the given index key. *Rebuildin an index never has an influence on the Clustering Factor, but instead it requires the respective table should be re-organized.*

The impact of rebuilding an index can be quite significant, for instance:

- *analyze index … validate structure;*
  This is a valid method to inspect the index, but it gets an exclusive table lock while analyzing the index. Eventhough you can use *online* switch, but the *Index_Stats* dynamic table might not be pupolated. For large indexes, this can have significant influence on the Database Performance during peak time.
- Redo Generation is another factor that the peformance might be impacted.
- An **index coalesce** is often preferred method instead of an index rebuild. It has the following advantages:
  - It does not require approximately 2 times the disk space.
  - It can run always online
  - It does not restructure the index.

****

<< Solution of determining when an index should be rebuilt>>

1- Finding indexes that have height **(Blevel + 1) > 4**:

```
SQL> Select owner, index_name, table_name, blevel, (SELECT num_rows FROM
dba_tables WHERE owner = idx.owner AND table_name = idx.table_name)  tlb_num_rows
FROM dba_indexes idx  WHERE idx.owner = 'owner_name' AND idx.blevel > 3;
```

2- Then analyze indexes to find the ratio of (DEL_LF_ROWS/LF_ROWS*100) is > 20:

```
SQL> analyze index I_RECID validate structure; -- First analyze the suspect index

SQL> SELECT name, height, lf_rows, del_lf_rows, round((del_lf_rows/lf_rows)*100,1)
as ratio FROM INDEX_STATS;

NAME       HEIGHT   LF_ROWS  DEL_LF_ROWS   RATIO
---------- ---------- ---------- ---------------- -------
I_RECID         5       1614265     1529545          94.8
```

- As already discussed, the Blevel is not always an indication for a bad index. Please check Oracle Support documents. However, rebuilding an index may help performance in specific cases.

3- You can see that height of candidate index is **5** and also high number of DEL_LF_ROWS with Ratio about 94.8%.
You can check execution plan with its statistics to see the usage of the index and also ***consistent gets and phyisical reads***.

4- Now you can rebuild the candidate index:

```
SQL> alter index I_RECID rebuild online;
```

5- Finally, you can redo steps 2-4 to compare with the previous statistics. After select from INDEX_STATS you should see the values of **DEL_LF_ROWS and RATIO are zero**.

6- Finally, the height of index is reduced below 4 and **DEL_LF_ROWS is 0.** Then you can check the query perfomance after rebuilding the index in that particular situation.

- ***References: Oracle Support and Oracle Performance Tuning Docs***

<<Hassan Irani – 04.06.2025>>