
Identifying Which Indexes to Rebuild

By
Don Burleson
Editor-in-Chief, Oracle Internals

Corporate Headquarters	EMEA Headquarters
Embarcadero	Embarcadero Technologies,
Technologies	Ltd.
100 California Street	Thames House
12th Floor	17 Marlow Road
San Francisco, CA 94111	Maidenhead
www.embarcadero.com	Berkshire
	SL6 7AA
	United Kingdom
	www.embarcadero.co.uk

Table of Contents

Identifying Which Indexes to Rebuild

What's Going on Here?	3
BIO.....	8
About Embarcadero Technologies	8

What's Going on Here?

Indexes require rebuilding when deleted leaf nodes appear or when the index has spawned into too many levels of depth. While it is tempting to write a script that rebuilds every index in the SAPR3 schema, bear in mind that SAP contains many thousands of indexes, and a complete rebuild can be very time consuming. Hence, we need to develop a method to identify those indexes that will get improved performance with a rebuild. Let's look at a method for accomplishing this task. As I mentioned earlier in this chapter, Oracle index nodes are not physically deleted when table rows are deleted, nor are the entries removed from the index. Rather, Oracle "logically" deletes the index entry and leaves "dead" nodes in the index tree. These deleted leaf nodes can be easily identified by running the IDL.SQL script (shown in Listing 5-12); the output from the script is shown in Listing 5-11.

Index	# of rep keys	dist. keys	# deleted leaf rows	Height	blk gets per access
I_POS	1	477,705	711	3	4
I_POS_QTY	96,680	11,576	4,614	3	24

Listing 5-11: A detailed Index Report

In Listing 5-11 we see several important statistics.

The number of deleted leaf nodes

The term "deleted leaf node" refers to the number of index nodes that have been logically deleted as a result of row deletes. Remember that Oracle leaves "dead" index nodes in the index when rows are deleted. This is done to speed up SQL deletes, since Oracle does not have to rebalance the index tree when rows are deleted. Index height

The height of the index refers to the number of levels that are spawned by the index as a result in row inserts. When a large amount of rows are added to a table, Oracle may spawn additional levels of an index to accommodate the new rows. Hence, an Oracle index may have four levels, but only in those areas of the index tree where the massive inserts have occurred. Oracle indexes can support many millions of entries in three levels, and any SAP index that has four or more levels would benefit from rebuilding. Gets per index access

The number of "gets" per access refers to the amount of logical I/O that is required to fetch a row with the index. As you may know, a logical "get" is not necessarily a physical I/O since much of the index may reside in the Oracle buffer cache. However, any SAP index with a number greater than 10 would probably benefit from an index rebuild. Unfortunately, Oracle does not make it easy to capture this information. In Oracle we must issue these commands:

```
ANALYZE INDEX index_name COMPUTE STATISTICS
ANALYZE INDEX index_name VALIDATE STRUCTURE
```

After you analyze the report above, you may want to consider rebuilding any index where the height is more than three levels, since three levels will support millions of index entries. Note that

Identifying Which Indexes to Rebuild

Oracle indexes will spawn to a fourth level only in areas of the index where a massive insert has occurred, such that 99% of the index has three levels, but the index is reported as having four levels.

We might want to rebuild an index if the block gets per access is greater than five, since excessive blocks gets indicate a fragmented b-tree structure. Another rebuild condition would be cases where deleted leaf nodes comprise more than 20% of the index nodes. As you may know, you can easily rebuild an Oracle index with the command:

```
ALTER INDEX index_name REBUILD
```

Finally, we may want to change a regular Oracle index to a bitmapped index. As you may know, bitmapped indexes are faster when the index key has less than 25 distinct values. If you have an index where the number of distinct keys is less than 25, you are on release 7.3.4 or above of Oracle, and you also get permission from SAP-AG, you are free to change index structures.

The SQL that created this report is shown in Listing 5-12.

```
revsp2t> cat id1.sql
set pages 9999;
set heading off;
set feedback off;
set echo off;

spool id4.sql;

select '@id2.sql' from dual;

select 'analyze index '||owner||'.'||index_name||' validate structure;',
 '@id3.sql;'
from dba_indexes
where
owner not in ('SYS','SYSTEM');

spool off;

set heading on;
set feedback on;
set echo on;

@id4.sql
@id5.sql

revsp2t> cat id2.sql
create table temp_stats as
select
name ,
most_repeated_key ,
distinct_keys ,
del_if_rows ,
```

Identifying Which Indexes to Rebuild

```
height ,
blks_gets_per_access
from index_stats;
```

```
revsp2t> cat id3.sql
insert into temp_stats
(select
name ,
most_repeated_key ,
distinct_keys ,
del_lf_rows ,
height ,
blks_gets_per_access
from index_stats
);
```

```
id5.sql
Rem ind_fix.sql - Shows the details for index stats
```

```
set pagesize 60;
set linesize 100;
set echo off;
set feedback off;
set heading off;
```

```
column c1 format a18;
column c2 format 9,999,999;
column c3 format 9,999,999;
column c4 format 999,999;
column c5 format 99,999;
column c6 format 9,999;
```

```
spool idx_report.lst;
```

```
prompt
prompt
```

```
prompt '          # rep dist. # deleted      blk gets
prompt Index      keys  keys leaf rows Height per access
prompt -----
```

```
select distinct
name c1,
most_repeated_key c2,
distinct_keys c3,
del_lf_Rows c4,
height c5,
blks_gets_per_access c6
from temp_stats
```

Identifying Which Indexes to Rebuild

```
where
height > 3
or
del_lf_rows > 10
order by name;

spool off;

spool id6.sql;

select 'alter index '||owner||'.'||name||' rebuild tablespace '||tablespace_name
||';'
from temp_stats, dba_indexes
where
temp_stats.name = dba_indexes.index_name
and
(height > 3
or
del_lf_rows > 10);v

select 'analyze index '||owner||'.'||name||' compute statistics;'
from temp_stats, dba_indexes
where
temp_stats.name = dba_indexes.index_name
and
(height > 3
or
del_lf_rows > 10);

spool off;
```

Listing 5-12. The id1.sql and other scripts to produce a detailed index report

Now that we have identified the candidates for an index rebuild, we can run the following script during SAP system downtime to re-build all of the indexes (see Listing 5-13).

```
Set heading off;
Set pages 9999;
Spool run_rebuild.sql;

select 'alter index sapr3.'||
index_name||
' rebuild tablespace '||
tablespace_name||';'
from dba_indexes
where owner = SAPR3.;
```

```
spool off;
@run_rebuild
Listing 5-13. A script to generate the index rebuild syntax
```

Using ALTER INDEX REBUILD to Rebuild Indexes

The ALTER INDEX index_name REBUILD command is very safe way to rebuild indexes. Here is the syntax of the command:

```
alter index index_name
rebuild
tablespace tablespace_name
storage (initial new_initial next new_next freelists new_freelist_number )
```

Unlike the traditional method where we drop the index and recreate it, the REBUILD command does not require a full table scan of the table, and the subsequent sorting of the keys and rowids. Rather, the REBUILD command will perform the following steps:

1. Walk the existing index to get the index keys.
2. Populate temporary segments with the new tree structure.
3. Once the operation has completed successfully, drop the old tree, and rename the temporary segments to the new index.

As you can see from the steps, you can rebuild indexes without worrying that you will accidentally lose the index. If the index cannot be rebuilt for any reason, Oracle will abort the operation and leave the existing index intact. Only after the entire index has been rebuilt does Oracle transfer the index to the new b-tree. Most Oracle administrators run this script, and then select the index that they would like to rebuild. Note that the TABLESPACE clause should always be used with the ALTER INDEX REBUILD command to ensure that the index is rebuilt in the same tablespace as the existing index. Be aware that you must have enough room in that tablespace to hold all of the temporary segments required for the index rebuild, so most SAP Oracle administrators will double-size index tablespaces with enough space for two full index trees.

**This article is an excerpt from Don Burleson's latest book, "Oracle/SAP Administration," by O'Reilly & Associates

BIO

Don Burleson specializes in Oracle technology, tuning, and data warehousing. The author of eight books on database management, and more than 50 articles in national magazines, Don has more than 17 years experience as a database administrator, working with some of the worlds most sophisticated Oracle systems. Don is a frequent contributor to Oracle Internals magazine, and serves as editor-in-chief of Oracle Internals Magazine. He can be reached at burleson@frontiernet.net.

About Embarcadero Technologies

Embarcadero Technologies, Inc. is a leading provider of strategic data management solutions that help companies to improve the availability, integrity, accessibility, and security of corporate data. Nearly 12,000 customers, including 97 of the Fortune 100, rely on Embarcadero Technologies solutions to maximize their return on corporate data assets and to meet the challenges of explosive data growth, escalating data security requirements, and complex, multi-platform data environments.

© 2006, Embarcadero Technologies, Inc. Embarcadero, the Embarcadero Technologies logos and all other Embarcadero Technologies product or service names are trademarks of Embarcadero Technologies, Inc. All other trademarks are property of their respective owners.