



## Why is my Oracle10g Database SLOWER than my Oracle9i Database?

This paper illustrates performance issues I've discovered with some of the features of the Oracle10g database..

This material contains excerpts from my 2-day Advanced Tuning course.

### Index Monitoring

Extra indexes are not only used but can significantly slow down batch processing. Oracle will create before images and journaling entries for each update or delete to a table. Oracle also does this for each index affected by the DML operation.

Oracle9i allows for index usage to be easily monitored. Simply issue 'alter index <index name> MONITORING USAGE' (NOMONITORING USAGE turns it back off) to start the monitoring process. Monitor V\$\_OBJECT\_USAGE, in particular, the USED column will be set to 'YES' when the index appears in an explain plan.

Column	Data Type	Null?	Updatable	Comments
INDEX_NAME	VARCHAR2 (30)	N		Name of the index
TABLE_NAME	VARCHAR2 (30)	N		Name of the table upon which the ind...
MONITORING	VARCHAR2 (3)	Y		Whether the monitoring feature is on
USED	VARCHAR2 (3)	Y		Whether the index has been accessed
START_MONITORING	VARCHAR2 (19)	Y		When the monitoring feature is turned...
END_MONITORING	VARCHAR2 (19)	Y		When the monitoring feature is turned...

The spool the output from this script: 'select 'alter index ' || index\_name || ' monitoring usage; from user\_indexes' can be used to trip this monitoring on.

Run a script to check for index usage and drop the unused indexes! Again, a script can be used to drop the unused indexes. Ask Dan for this script.

## Things I've Heard...



**❖ Things I've Heard...**

- SQL with Histograms not working correctly
- SQL with Different Explain Plans for no apparent reason
- Try COMPATIBLE setting
  - Set to 9.2.0.0.0
  - ALTER SESSION SET COMPATIBLE = 9.2.0.0.0
  - INIT.ora sessing

---

[www.DanHotka.com](http://www.DanHotka.com)

## Index Clustering Factor

Index Clustering factor is the relationship between the Index leaf block and how many different data blocks each leaf block points to. The CBO uses this ratio to determine if it is more efficient to do a full table scan versus a range scan when being asked to perform a range scan.

The example in the slide shows two tables (with permission from Jonathan Lewis). TEST\_TABLE1 and TEST\_TABLE2 have the same attributes. The data is in significantly different order.

# Index-Clustering Factor

```
TOAD - [USER0@ORAXP91 SQL Editor (select * from Test_Table1 where col1 = 45)]
File Edit Grid SQL-Window Create Database Tools View DBA Debug Team Coding Window Help
SQL
select * from Test_Table1 where col1 = 45:
Data Explain Plan Auto Trace DBMS Output Code Statistics Script Output
Operation Object Name Rows Bytes Cost Object Node
SELECT STATEMENT Optimizer Mode=CHOOSE 30 210 2
  TABLE ACCESS BY INDEX ROWID TEST_TABLE1 30 210 2
    INDEX RANGE SCAN TEST_TABLE1_IDX 30 1
```

Test Table 1 performs a nice range scan.

# Index-Clustering Factor

```
TOAD - [USER0@ORAXP91 SQL Editor (select * from Test_Table2 where col1 = 45)]
File Edit Grid SQL-Window Create Database Tools View DBA Debug Team Coding Window Help
SQL
select * from Test_Table2 where col1 = 45:
Data Explain Plan Auto Trace DBMS Output Code Statistics Script Output
Operation Object Name Rows Bytes Cost Object Node
SELECT STATEMENT Optimizer Mode=CHOOSE 30 3K 11
  TABLE ACCESS FULL TEST_TABLE2 30 3K 11
```

Test Table 2 performs a full table scan at a much higher cost.

# Index Clustering Factor

TOAD - [USER@ORA10P91 SQL Editor (Part1\_SQL5.sql)]

```

select /*+ index(test_table2) */ * from Test_Table2 where col1 =

```

Operation	Object Name	Rows	Bytes	Cost	Object
SELECT STATEMENT Optimizer Mode=CHOOSE		30		31	
TABLE ACCESS BY INDEX ROWID	TEST_TABLE2	30	3K	31	
INDEX RANGE SCAN	TEST_TABLE2_IDX	30		1	

1

Test Table 2 can do a range scan with a hint but at 3 times the cost.

# Index\_Info.sql

```

SELECT i.table_name, i.index_name, t.num_rows, t.blocks, i.avg_data_blocks_per_key,
i.avg_leaf_blocks_per_key, i.clustering_factor, o.created
from user_indexes i, user_objects o, user_tables t
where i.index_name = o.object_name
and i.table_name = t.table_name

```

TABLE_NAME	INDEX_NAME	NUM_ROWS	BLOCKS	AVG_DATA	AVG_LEAF	CLUSTERING_
A	A_STATUS_IDX	1000	8	3	3	3
B	B_STATUS_IDX	100	4	1	1	1
C	C_B_ID_IDX	1000	8	3	1	300
DEPT_CHAR	DEPT_CHAR_PRIMARY_					
DEPT	DEPT_PRIMARY_KEY					
MASTER	MASTER_LASTNAME					
B	FK_B_ID	100	4	1	1	1
TEAMS	TEAMS_PK					
TEST_TABLE1	TEST_TABLE1_IDX	6000	101	1	1	92
TEST_TABLE2	TEST_TABLE2_IDX	6000	101	30	1	6000
TOAD_PLAN_TABLE	TPTBL_IDX					

www.DanHotka.com

INDEX\_INFO shows what is going on. Notice the clustering factor of Test Table 2 is very high. The clustering Factor of Test Table1 is low.

The bottom line is the more the data is in the order of the index key, the more Oracle will use it on range scans.



## Automatic Space Storage Management

- ❖ **Only available with Locally-Managed TS**
  - Oracle9v2+
  - AUTO – uses Automatic Space Storage Mngt
  - Manual – uses freelists (existing technology)

```
CREATE TABLESPACE my_tablespace DATAFILE '/oracle/data/myts01.dbf'  
SIZE 500M  
EXTENT MANAGEMENT LOCAL  
SEGMENT SPACE MANAGEMENT AUTO;
```

---

[www.DanHotka.com](http://www.DanHotka.com)

# Automatic Space Storage Management

❖ How it works:

The diagram illustrates the Automatic Space Management process. It shows a sequence of blocks in a segment extent, labeled from left to right as BB=1, BB=1, BB=1, BB=1, BB=3, BB=2, BB=2, BB=2, BB=2, BB=2, BB=2, BB=0, BB=0, BB=0. A green vertical line marks the 'Low HWM' (High Water Mark) at the fourth block (BB=1). A blue vertical line marks the 'High HWM' at the tenth block (BB=2). A green oval labeled 'Inserts in round-robin order' with arrows indicates that new data is inserted into blocks starting from the Low HWM and moving towards the High HWM in a round-robin fashion. A red arrow labeled 'Unformatted' points to the right, indicating that blocks to the right of the High HWM are not yet formatted.

www.DanHotka.com

Automatic Space Management is a new tablespace feature for Oracle10g. If used, Oracle does not format the whole data file but just 6 blocks at a time. Oracle will then round-robin insert into these blocks when doing inserts. Doing an insert into a new block will always give a higher clustering factor than just using a regular tablespace.

# Index\_Info.sql

```

SELECT i.table_name, i.index_name, t.usa_rows, t.blocks, i.avg_data_blocks_per_key,
i.avg_leaf_blocks_per_key, i.clustering_factor,
to_char(i.created, 'MM/DD/YYYY HH24:MI:SSSSS') Created
from user_indexes i, user_objects o, user_tables t
where i.index_name = o.object_name
and i.table_name = t.table_name
order by 1;

```

TABLE_NAME	INDEX_NAME	NUM_ROWS	BLOCKS	AVG_DATA_BLOCKS_PER_KEY	AVG_LEAF_BLOCKS_PER_KEY	CLUSTERING_FACTOR
TEST_TABLE1	TEST_TABLE1_IDX	6000	101	1	1	1022

www.DanHotka.com

This example shows Test Table 1 against an Oracle10g tablespace using ASM. Notice the higher clustering factor.

ASM is good for RAC and applications with lots of insert activity but possibly at the cost of using indexes on range scan.

## **Case Study**

Extra indexes are not only used but can significantly slow down batch processing. Oracle will create before

## **Summary**

This article fully describes several things to watch and monitor with your Oracle10g database. Clustering factor and the Automated Segment Management feature could be an issue. Too many indexes can adversely affect any Oracle database. Compatibility might help. The case study illustrates adding a lot more memory to the Oracle10g database than what the Oracle9i database required.

Dan Hotka is a Training Specialist who has over 29 years in the computer industry and over 24 years of experience with Oracle products. He is an internationally recognized Oracle expert with Oracle experience dating back to the Oracle V4.0 days. Dan's latest book is the [SQL Developer Handbook](#) by Oracle Press. He is also the author of [Oracle9i Development By Example](#) and [Oracle8i from Scratch](#) by Que and has co-authored 7 other popular books including the [Database Oracle10g Linux Administration](#) by Oracle Press. He is frequently published in Oracle trade journals, and regularly speaks at Oracle conferences and user groups around the world. Visit his website at [www.DanHotka.com](http://www.DanHotka.com). Dan can be reached at [dhotka@earthlink.net](mailto:dhotka@earthlink.net).

Dan Hotka - Author/Instructor/Expert  
[www.DanHotka.com](http://www.DanHotka.com)  
[DHotka@Earthlink.net](mailto:DHotka@Earthlink.net)  
515 279-3361