CSCI317 – Final Project

Student Name: Trang Nguyen

Student Number: 6166994 – Username: tttn941

Solution 2:

(1) a short description of an improvement

ASUMPTIONS: Every assumption in solution1.pdf applies to this task.

As the View is correctly merged in the SELECT statement, to speed up the query in task2.sql, a B*tree index on LINEITEM(L_ORDERKEY, L_COMMITDATE, L_RECEIPTDATE) has been created.

As there was no significant improvement in terms of the number of reads of the indexed query compared to the query in task2.sql, I've decided to cache the B*tree index so that the number of physical reads can be reduced.

To do so, I've followed the following steps:

1. I've allocated 60MB (as the index is as large as 60MB) to the 32K Buffer.
2. Then, I've created a tablespace with the size of 60MB (the size of the index) and a block size of 32K, as B-tree indexes should use the largest supported size of data blocks (32K) [1].
3. Then I used Oracle alter table and alter index commands to move tables and indexes into these new tablespaces.

(2) information about the benefits from an improvement

The number of physical reads is significantly improved compared to the query in task2.sql, as shown in the figures below:

```
SQL> set lines 196;
SQL> column table_space format a48 trunc;
SQL> column file_name   format a48 trunc;
SQL> set numwidth 10;
SQL> Rem I/O should be spread evenly accross drives. A big difference between
SQL> Rem phys_reads and phys_blks_rd implies table scans are going on.
SQL> select table_space, file_name,
  2         phys_reads reads, phys_blks_rd blks_read, phys_rd_time read_time,
  3         phys_writes writes, phys_blks_wr blks_wrt, phys_wrt_tim write_time,
  4         megabytes_size megabytes,
  5         round(decode(phys_blks_rd,0,0,phys_rd_time/phys_blks_rd),2) avg_rt,
  6         round(decode(phys_reads,0,0,phys_blks_rd/phys_reads),2) "blocks/rd"
  7   from stats$files order by table_space, file_name;

TABLE_SPACE                                FILE_NAME                                          READS  BLKS_READ  READ_TIME     WRITES   BLKS_WRT WRITE_TIME  MEGABYTES     AVG_RT  blocks/rd
------------------------------------------ -------------------------------------------------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
SYSAUX                                     /opt/oracle/oradata/DB/sysaux01.dbf                    10         13          1          2          2          1        881        .08        1.3
SYSTEM                                     /opt/oracle/oradata/DB/system01.dbf                    97        105         10         55        106        152        965         .1       1.08
TPCHR                                      /opt/oracle/oradata/DB/tpchr.dbf                    10020      39573        259          0          0          0       3146        .01       3.95
UNDOTBS1                                   /opt/oracle/oradata/DB/undotbs01.dbf                    9          9          1         19         42         25        624        .11          1
USERS                                      /opt/oracle/oradata/DB/users01.dbf                      0          0          0          0          0          0         31          0          0
```

The number of physical reads and physical blks_rd *before* the index being cached

```
SQL> set lines 196;
SQL> column table_space format a48 trunc;
SQL> column file_name   format a48 trunc;
SQL> set numwidth 10;
SQL> Rem I/O should be spread evenly accross drives. A big difference between
SQL> Rem phys_reads and phys_blks_rd implies table scans are going on.
SQL> select table_space, file_name,
  2         phys_reads reads, phys_blks_rd blks_read, phys_rd_time read_time,
  3         phys_writes writes, phys_blks_wr blks_wrt, phys_wrt_tim write_time,
  4         megabytes_size megabytes,
  5         round(decode(phys_blks_rd,0,0,phys_rd_time/phys_blks_rd),2) avg_rt,
  6         round(decode(phys_reads,0,0,phys_blks_rd/phys_reads),2) "blocks/rd"
  7   from stats$files order by table_space, file_name;

TABLE_SPACE                                FILE_NAME                                          READS  BLKS_READ  READ_TIME     WRITES   BLKS_WRT WRITE_TIME  MEGABYTES     AVG_RT  blocks/rd
------------------------------------------ -------------------------------------------------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------
SYSAUX                                     /opt/oracle/oradata/DB/sysaux01.dbf                     3          3          0          0          0          0        881          0          1
SYSTEM                                     /opt/oracle/oradata/DB/system01.dbf                    15         15          2          0          0          0        965        .13          1
TASK2_INDEX_32K                            /opt/oracle/oradata/DB/task2_32k_tbs.dbf             1839       1839         15          0          0          0         63        .01          1
TPCHR                                      /opt/oracle/oradata/DB/tpchr.dbf                     1984       7255         46          0          0          0       3146        .01       3.66
UNDOTBS1                                   /opt/oracle/oradata/DB/undotbs01.dbf                    0          0          0          0          0          0        624          0          0
USERS                                      /opt/oracle/oradata/DB/users01.dbf                      0          0          0          0          0          0         31          0          0

6 rows selected.
```

The number of physical reads and physical blks_rd *after* the index being cached

	NAME	VALUE
1	db block gets from cache	101084
2	consistent gets from cache	12074123
3	physical reads cache	285092

Statistics from V$SYSSTAT

“The buffer cache hit ratio calculates how often a requested block has been found in the buffer cache without requiring disk access” [2].

In this case, the system is an OnLine Transaction Processing (OLTP) system with many short transactions, the system Hit Ratio should ideally be >= 95%.

In this case, based on V$SYSSTAT, our Hit Ratio is:

```
Hit Ratio = 1 - (('physical reads cache')/('consistent gets from cache' + 'db block gets from cache'))
          = ~97.66%
```

Thus, it can be considered as a well tunned system.

(3) information about the costs of an improvement, i.e. documented investments into transient and persistent storage

- In terms of transient storage, I've invested 60MB into the 32K Buffer Cache.
- In terms of persistent storage, I've invested 60MB into creating a new tablespace.

```
ALTER SYSTEM SET db_32K_cache_size = 60M SCOPE=SPFILE;


CREATE TABLESPACE TASK2_INDEX_32K
BLOCKSIZE 32K
DATAFILE '/opt/oracle/oradata/DB/task2_32k_tbs.dbf'
SIZE 60M AUTOEXTEND ON
extent management local;
```

Statements for allocating memory to 32K buffer and the tablespace

```
SQL> set feedback on
NAME                 TYPE        VALUE
-------------------- ----------- -----
db_32k_cache_size    big integer 60M
```

Size of the 32K buffer

(4) a report from implementation of an improvement

```
SQL> /* Create the B*tree index */

SQL> CREATE INDEX L_IDX ON LINEITEM(l_orderkey, l_commitdate, l_receiptdate) COMPRESS 1;


Index L_IDX created.
```

```
SQL> /* Allocate memory to 32K buffer cache */
SQL> ALTER SYSTEM SET db_32K_cache_size = 60M SCOPE=SPFILE;

System SET altered.

SQL> CREATE TABLESPACE TASK2_INDEX_32K
  2  BLOCKSIZE 32K
  3  DATAFILE '/opt/oracle/oradata/DB/task2_32k_tbs.dbf'
  4  SIZE 60M AUTOEXTEND ON
  5  extent management local;

TABLESPACE TASK2_INDEX_32K created.

SQL> ALTER USER TPCHR QUOTA 60M ON TASK2_INDEX_32K;

User TPCHR altered.

SQL> ALTER INDEX TPCHR.L_IDX REBUILD TABLESPACE TASK2_INDEX_32K;

Index TPCHR.L_IDX altered.
```

```
SQL> /* CONNECT AS SYSTEM IN SQLPLUS AND RUN @utlbstat */
SQL> select
  2    o_orderpriority,
  3    count(*) as order_count
  4  from
  5    orders
  6  where
  7   o_orderdate >= '28-JUN-92'
  8   and o_orderdate <= '18-SEP-98'
  9   and exists (
  10    select
  11      *
  12    from
  13      lineitem
  14    where
  15      l_orderkey = o_orderkey
  16      and l_commitdate < l_receiptdate
  17  )
  18   group by
  19    o_orderpriority
  20   order by
```

```
  21    o_orderpriority;

  O_ORDERPRIORITY     ORDER_COUNT
  ------------------  -----------
  1 - URGENT                76729
  2 - HIGH                  76473
  3 - MEDIUM                75810
  4 - NOT SPECIFIED         76405
  5 - LOW                   76567

  5 rows selected.

SQL> /* CONNECT AS SYSTEM IN SQLPLUS AND RUN @utlestat */
SQL> DROP TABLESPACE TASK2_INDEX_32K INCLUDING CONTENTS AND DATAFILES;

TABLESPACE TASK2_INDEX_32K dropped.

SQL> DROP INDEX L_IDX;

Index L_IDX dropped.
```

Reference

[1] Presentation 34 – Performance Tuning of Relational Database Server (1), page 20.

[2] Docs.oracle.com. 2021. *Tuning the Database Buffer Cache*. [online] Available at: <https://docs.oracle.com/database/121/TGDBA/tune_buffer_cache.htm#TGDBA533> [Accessed 19 June 2021].