OLTP Compression Approach
Oracle FLEXCUBE Universal Banking
Release 14.3.0.0.0
[May] [2019]
# Table of Contents

1. **INTRODUCTION** ......................................................................................................................... 1-1
   1.1 Introduction to Oracle 11g Advanced OLTP Table Compression .............................................. 1-1
   1.2 Scope ........................................................................................................................................... 1-1
   1.3 Advantages ................................................................................................................................. 1-1

2. **REQUIREMENT OR PROBLEM STATEMENT** ............................................................................... 2-1

3. **PREREQUISITES** ......................................................................................................................... 3-1
   3.1 Software Required .................................................................................................................... 3-1

4. **ORACLE ADVANCED OLTP COMPRESSION DESCRIPTION** ....................................................... 4-1
   4.1 Core Features ............................................................................................................................ 4-1
   4.2 Threshold Based Block Compression ......................................................................................... 4-2
   4.3 How to Enable Table Compression ........................................................................................... 4-3
   4.4 Licensing ................................................................................................................................. 4-3
   4.5 Approach .................................................................................................................................... 4-3
      4.5.1 Data Preparation ................................................................................................................ 4-3
      4.5.2 Test Cases .......................................................................................................................... 4-3
   4.6 Environment Used ..................................................................................................................... 4-4
   4.7 Comparison Measurement Results .......................................................................................... 4-4
      4.7.1 Storage Consumed .............................................................................................................. 4-4
      4.7.2 Performance with Inserts ................................................................................................ 4-4
      4.7.3 Performance with Updates ................................................................................................ 4-4
      4.7.4 Performance with Deletes ................................................................................................ 4-4
      4.7.5 Performance with Selects ................................................................................................ 4-4
      4.7.6 Performance with Statistics Computation ....................................................................... 4-4
      4.7.7 Performance with Index Creation ...................................................................................... 4-4
      4.7.8 Disk Reads .......................................................................................................................... 4-4
1. **Introduction**

1.1 **Introduction to Oracle 11g Advanced OLTP Table Compression**

Oracle Advanced OLTP Table compression feature was introduced in version 11g and this feature allows data to be compressed during all DML operations. Oracle 11g uses a unique algorithm that eliminates duplicate values within a single block even across multiple columns. Similar to Oracle 9i, each block contains a symbol table that maintains compression metadata and this allows better performance compared to the global symbol table based algorithms.

To reduce the overhead during compression Oracle 11g uses a threshold based compression, which only compresses the data within a block once the threshold is reached, and thus eliminates the overhead to compress the data every time it is written.

1.2 **Scope**

This document brings out the feasibility of performing a table level compression using Oracle 11g Advanced OLTP table compression feature. This document also presents results of a test done using sample data with various operations and thus quantifies the benefits achieved in some of these operations.

1.3 **Advantages**

The benefits of using Oracle Advanced OLTP compression feature is much beyond just disk storage savings. One significant advantage is its ability to read compressed blocks directly without having to uncompress it. Therefore there is no measurable performance degradation during read operations on compressed tables.
2. Requirement or Problem Statement

The requirement is to display the capability of the Oracle OLTP table compression feature and to measure the performance with various operations such as Read, DML, Statistics collection etc.
3. Prerequisites

3.1 Software Required

Oracle Database Server 11g Release 1 (11.1.0.6)

Download and install the Oracle Database server 11g version 11.1.0.6 from the below link:

4. Oracle Advanced OLTP Compression Description

4.1 Core Features

Oracle has been one of the pioneers in introducing database compression technology. Oracle database 9i introduced Table Compression several years ago to allow data to be compressed during bulk load operations which was ideally suited for data warehouse environments.

Oracle database 11g introduced a new feature called OLTP table compression. This allows data to be compressed during all types of data manipulation operations including conventional DML such as INSERT or UPDATE.

Oracle database 11g uses a unique compression algorithm specifically designed to work with relational data. The algorithm works by eliminating duplicate values within a database block, even across multiple columns. Compressed blocks contain a structure called a symbol table that maintains compression metadata. When a block is compressed, duplicate values are eliminated by first adding a single copy of the duplicate value to the symbol table. Each duplicate value is then replaced by a short reference to the appropriate entry in the symbol table.

Through this innovative design, compressed data is self-contained within the database block as the metadata used to translate compressed data into its original state is contained within the block. When compared with competing compression algorithms that maintain a global database symbol table, Oracle’s unique approach offers significant performance benefits by not introducing additional I/O when accessing compressed data.

Compressed Block vs. Noncompressed Block

<table>
<thead>
<tr>
<th>Invoice_ID</th>
<th>Cust_Name</th>
<th>Cust_Addr</th>
<th>Sales_amt</th>
</tr>
</thead>
<tbody>
<tr>
<td>1233033</td>
<td>Meyer</td>
<td>11 Homestead Rd</td>
<td>13.99</td>
</tr>
<tr>
<td>1212300</td>
<td>Meyer</td>
<td>11 Homestead Rd</td>
<td>1.99</td>
</tr>
<tr>
<td>1243012</td>
<td>McGlyn</td>
<td>3 Main Street</td>
<td>1.99</td>
</tr>
<tr>
<td>9923032</td>
<td>McGlyn</td>
<td>3 Main Street</td>
<td>1.99</td>
</tr>
<tr>
<td>6633023</td>
<td>McGlyn</td>
<td>3 Main Street</td>
<td>1.99</td>
</tr>
<tr>
<td>2133056</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Not Compressed

Compressed

<table>
<thead>
<tr>
<th>Header Information</th>
<th>Symbol Table</th>
<th>Raw Data</th>
<th>Free Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>1233033 Meyer 11</td>
<td>1.99</td>
<td>1212300</td>
<td>9923032</td>
</tr>
<tr>
<td>Homestead Rd</td>
<td></td>
<td>1.99</td>
<td>3 Main Street</td>
</tr>
<tr>
<td>Meyer 11 Homestead Rd</td>
<td>1.99</td>
<td>3 Main Street</td>
<td></td>
</tr>
<tr>
<td>1243012 Meyer 11 Homestead Rd</td>
<td>1.99</td>
<td>3 Main Street</td>
<td></td>
</tr>
<tr>
<td>9923032 McGlyn 3 Main Street</td>
<td>1.99</td>
<td>3 Main Street</td>
<td></td>
</tr>
<tr>
<td>McGlyn 3 Main Street</td>
<td>1.99</td>
<td>3 Main Street</td>
<td></td>
</tr>
<tr>
<td>6633023 McGlyn 3 Main Street</td>
<td>1.99</td>
<td>3 Main Street</td>
<td></td>
</tr>
<tr>
<td>2133056</td>
<td></td>
<td>1.99</td>
<td>3 Main Street</td>
</tr>
</tbody>
</table>
4.2 **Threshold Based Block Compression**

With Oracle’s capability to read compressed blocks directly, there is no significant overhead in performance during read operations. However, compression requires additional work to be performed while writing the data making it unavoidable to eliminate performance overhead for write operations. To minimize this performance overhead, oracle compresses a block in batch mode rather than compressing data every single time a write operation takes place.

A newly initialized block remains uncompressed until data in the block reaches an internally controlled threshold. When a transaction causes the data in the block to reach this threshold, all contents of the block are compressed. Subsequently, as more data is added to the block and the threshold is again reached, the entire block is recompressed to achieve the highest level of compression. This process repeats until Oracle determines that the block can no longer benefit from further compression.

Only transactions that trigger the compression of the block will experience the compression overhead. Therefore, a majority of OLTP transactions on compressed blocks will have the exact same performance as they would with uncompressed blocks.
4.3 **How to Enable Table Compression**

The following example shows how OLTP Table compression can be enabled while table creation:

```sql
CREATE TABLE emp (
    Emp_id number,
    First_name varchar2 (20),
    Last_name varchar2 (20)
) COMPRESS FOR ALL OPERATIONS;
```

The following example shows how OLTP Table compression can be enabled for an existing table:

```sql
ALTER TABLE emp COMPRESS FOR ALL OPERATIONS;
```

Issuing the above statement will not compress the existing records. Only the new records written to this table will get compressed.

4.4 **Licensing**

Oracle Table compression feature introduced in Oracle Database 9i is a base feature of Enterprise Edition (EE) and continues to be so even in Oracle Database 11g. The new OLTP Table Compression feature, however, is a part of the Oracle Advanced Compression option that needs to be licensed in addition to the Enterprise Edition. Performance Measurement on OLTP Compressed Table – a sample

A Sample test was done for various operations on Compressed and Uncompressed tables and a performance comparison was done between the two.

4.5 **Approach**

4.5.1 **Data Preparation**

- A Compressed table created with structure similar to dba_objects.
- An Uncompressed table created with structure similar to dba_objects.
- An Uncompressed table created and populated with 10 million rows out of which 40 thousand rows were distinct. This table was used to populate the above 2 tables.

4.5.2 **Test Cases**

Comparison between the Compressed and Uncompressed table has been done for the following cases:-

- Storage consumed
- Performance with Inserts
- Performance with Updates
- Performance with Deletes
- Performance with Selects
- Performance with Statistics computation
4.6 **Environment Used**
- Operating System - Linux x86_64 2.6.18-53.el5
- Number of CPU’s - 8
- Physical Memory - 16GB

4.7 **Comparison Measurement Results**

4.7.1 **Storage Consumed**

Storage of compressed table was 48% lesser than uncompressed table.

4.7.2 **Performance with Inserts**

Inserts on compressed table were 3 times slower than uncompressed table.

4.7.3 **Performance with Updates**

Updates on compressed table were 8 times slower than uncompressed table.

4.7.4 **Performance with Deletes**

Deletes on compressed table were 15 percent faster than uncompressed table.

4.7.5 **Performance with Selects**

Select on compressed table took same time as uncompressed table.

4.7.6 **Performance with Statistics Computation**

Computation of Statistics on compressed table was 5 percent slower than uncompressed table.

4.7.7 **Performance with Index Creation**

Index creation on compressed table was 14 percent slower than uncompressed table.

4.7.8 **Disk Reads**

Disk reads on compressed table were 48 percent lesser than uncompressed table.