ORACLE

Getting Started with the Oracle Compression Advisor

April, 2022, Version 21.3

Public

Purpose statement

This document provides an overview of features and enhancements included in release Oracle Database 21c. It is intended solely to help you assess the business benefits of upgrading to Oracle Database 21c and to plan your I.T. projects.

Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Table of contents

Introduction

Oracle Advanced Compression includes a comprehensive set of compression capabilities to help organizations maximize resource utilization and reduce costs. It allows IT administrators to significantly reduce their overall database storage footprint, and improve performance, by enabling compression for all types of data, including:

Advanced Row Compression

Enables table data to be compressed during all types of data manipulation operations, including DML INSERT and UPDATE operations -- intelligent algorithm minimizes compression overhead during write operations, thereby making compression viable for both data warehouse and OLTP workloads.

Advanced LOB Compression

Provides compression for LOB segments managed by Oracle SecureFiles – a high performance and powerful infrastructure for managing unstructured data such as images, documents, videos and more.

Advanced Index Compression

Reduces the size of all supported unique and non-unique indexes- Advanced Index Compression HIGH automatically chooses the right compression per index block. Advanced Index Compression provides significant space savings while also improving performance for queries that are executed using indexes.

Compression Advisor

An easy way to get started, with Advanced Compression, is by using the free compression advisor. The "DBMS_COMPRESSION" PL/SQL package (commonly called compression advisor) gathers compression-related information within a database environment. This includes estimating the compressibility of both uncompressed partitioned, and non-partitioned tables, and gathering row-level compression information on previously compressed tables/partitions. Compression advisor provides organizations with the storage reduction information needed to make compression-related usage decisions.

The output of running compression advisor is an estimation of the compression ratio, for the specific table, that was the target of compression advisor. The output indicates the "COMPRESSION RATIO" presented as a number such as 2.1. This number indicates that, for this specific table or partition, the estimated compression ratio is 2.1x, which represents about a 50% reduction in the footprint of the table or partition should compression be enabled.

The compression ratio achieved in a given environment depends on the data being compressed, specifically the cardinality of the data. In general, organizations can expect to reduce their storage space consumption by a factor of 2x to 4x by using Advanced Row Compression. That is, the amount of space consumed by uncompressed data will be two to four times larger than that of the compressed data.

A version of compression advisor, which supports Oracle Database 9*i* Release 2 through 11g Release 1, is available on the Advanced Compression page on Oracle.com. This version can only report the compression ratio for data tables – those tables (and partitions) that would be targets for OLTP Table Compression.

Another version of the DBMS_COMPRESSION PL/SQL package is included with Oracle Database 11g Release 2 and above. This version can report the compression ratio for data tables (targets for Advanced Row

Compression), LOB segments managed by SecureFiles (targets for Advanced LOB Compression) and indexes (targets for Advanced Index Compression).

Compression Advisor is free to use with Oracle Database Enterprise Edition.

Using the GET_COMPRESSION_RATIO Procedure

When using the GET_COMPRESSION_RATIO procedure to estimate compression ratios, different constants are specified as parameters, these include:

Constant	Type	Value	Description
COMP_NOCOMPRESS	NUMBER	1	No compression
COMP_ADVANCED	NUMBER	2	Advanced row compression
COMP_QUERY_HIGH	NUMBER	4	High for query warehouse compression (Hybrid Columnar Compression)
COMP_QUERY_LOW	NUMBER	8	Low for query warehouse compression (Hybrid Columnar Compression)
COMP_ARCHIVE_HIGH	NUMBER	16	High archive compression (Hybrid Columnar Compression)
COMP_ARCHIVE_LOW	NUMBER	32	Low archive compression (Hybrid Columnar Compression)
COMP_BLOCK	NUMBER	64	Compressed block
COMP_LOB_HIGH	NUMBER	128	High compression level for LOB operations
COMP_LOB_MEDIUM	NUMBER	256	Medium compression level for LOB operations
COMP_LOB_LOW	NUMBER	512	Low compression level for LOB operations
COMP_INDEX_ADVANCED_HIGH	NUMBER	1024	High compression level for indexes
COMP_INDEX_ADVANCED_LOW	NUMBER	2048	Low compression level for indexes
COMP_RATIO_LOB_MINROWS	NUMBER	1000	Minimum required number of LOBs in the object for which LOB compression ratio is to be estimated
COMP_BASIC	NUMBER	4096	Basic table compression
COMP_RATIO_LOB_MAXROWS	NUMBER	5000	Maximum number of LOBs used to compute the LOB compression ratio
COMP_INMEMORY_NOCOMPRESS	NUMBER	8192	In-Memory with no compression
COMP_INMEMORY_DML	NUMBER	16384	In-Memory compression level for DML
COMP_INMEMORY_QUERY_LOW	NUMBER	32768	In-Memory compression level optimized for query performance
COMP_INMEMORY_QUERY_HIGH	NUMBER	65536	In-Memory compression level optimized on query performance as well as space saving
COMP_INMEMORY_CAPACITY_LOW	NUMBER	131072	In-Memory low compression level optimizing for capacity
COMP_INMEMORY_CAPACITY_HIGH	NUMBER	262144	In-Memory high compression level optimizing for capacity
COMP_RATIO_MINROWS	NUMBER	1000000	Minimum required number of rows in the object for which HCC ratio is to be estimated
COMP_RATIO_ALLROWS	NUMBER	-1	To indicate the use of all the rows in the object to estimate HCC ratio
OBJTYPE_TABLE	PLS_INTEGER	1	Identifies the object whose compression ratio is estimated as of type table
OBJTYPE_INDEX	PLS_INTEGER	2	Identifies the object whose compression ratio is estimated as of type index

GET_COMPRESSION_RATIO Procedure Parameters

Parameter	Description
scratchtbsname	Temporary scratch tablespace that can be used for analysis
ownname / tabowner	Schema of the table to analyze
tabname	Name of the table to analyze
objname	Name of the object
subobjname	Name of the partition or sub-partition of the object
comptype	Compression types for which analysis should be performed When the object is an index, only the following compression types are valid: COMP_INDEX_ADVANCED_HIGH (value 1024) and COMP_INDEX_ADVANCED_LOW (value 2048). **Note:** The following compression types cannot be specified in this parameter for any type of object: COMP_BLOCK (value 64) and COMP_BASIC (value 4096).
blkcnt_cmp	Number of blocks used by compressed sample of the table
blkcnt_uncmp	Number of blocks used by uncompressed sample of the table
row_cmp	Number of rows in a block in compressed sample of the table
row_uncmp	Number of rows in a block in uncompressed sample of the table
cmp_ratio	Compression ratio, blkcnt_uncmp divided by blkcnt_cmp
comptype_str	String describing the compression type
subset_numrows	Number of rows sampled to estimate compression ratio.
objtype	Type of the object, either OBJTYPE_TABLE or OBJTYPE_INDEX
lobname	Name of the LOB column
partname	In case of partitioned tables, the related partition name
lobcnt	Number of lobs actually sampled to estimate compression ratio
index_cr	List of indexes and their estimated compression ratios

Please see the Oracle Database documentation for additional information.

Usage Examples

Below are syntax examples, of the GET_COMPRESSION_RATIO procedure, to estimate the compression ratio of a data table, index and LOB.

Syntax for GET_COMPRESSION_RATIO for a data table and indexes:

```
DBMS_COMPRESSION.GET_COMPRESSION_RATIO (
 scratchtbsname        IN     VARCHAR2,
 ownname               IN     VARCHAR2,
 objname               IN     VARCHAR2,
 subobjname            IN     VARCHAR2,
 comptype              IN     NUMBER,
 blkcnt_cmp            OUT    PLS_INTEGER,
 blkcnt_uncmp          OUT    PLS_INTEGER,
 row_cmp               OUT    PLS_INTEGER,
 row_uncmp             OUT    PLS_INTEGER,
 cmp_ratio             OUT    NUMBER,
 comptype_str          OUT    VARCHAR2,
 subset_numrows        IN     NUMBER  DEFAULT COMP_RATIO_MINROWS,
 objtype               IN     PLS_INTEGER DEFAULT OBJTYPE_TABLE);
```

Example: Estimating Compression Ratio for Advanced Row Compression

```
SET SERVEROUTPUT ON
DECLARE
  l_blkcnt_cmp          PLS_INTEGER;
  l_blkcnt_uncmp        PLS_INTEGER;
  l_row_cmp             PLS_INTEGER;
  l_row_uncmp           PLS_INTEGER;
  l_cmp_ratio           NUMBER;
  l_comptype_str        VARCHAR2(32767);
BEGIN
  DBMS_COMPRESSION.GET_COMPRESSION_RATIO (
        scratchtbsname => 'USERS' ,
        ownname        => 'TEST' ,
        objname        => 'SALES' ,
        subobjname     =>  NULL ,
        comptype       =>  DBMS_COMPRESSION.COMP_ADVANCED,
        blkcnt_cmp     => l_blkcnt_cmp,
        blkcnt_uncmp   => l_blkcnt_uncmp,
        row_cmp        => l_row_cmp,
        row_uncmp      => l_row_uncmp,
        cmp_ratio      => l_cmp_ratio,
        comptype_str   => l_comptype_str,
        subset_numrows=> DBMS_COMPRESSION.comp_ratio_minrows,
        objtype        => DBMS_COMPRESSION.objtype_table
);
DBMS_OUTPUT.put_line( 'Number of blocks used by the compressed sample of the object          : ' ||
l_blkcnt_cmp);
DBMS_OUTPUT.put_line( 'Number of blocks used by the uncompressed sample of the object        : ' ||
l_blkcnt_uncmp);
DBMS_OUTPUT.put_line( 'Number of rows in a block in compressed sample of the object          : ' ||
l_row_cmp);
DBMS_OUTPUT.put_line( 'Number of rows in a block in uncompressed sample of the object        : ' ||
l_row_uncmp);
DBMS_OUTPUT.put_line ('Estimated Compression Ratio of Sample                                 : ' ||
l_cmp_ratio);
DBMS_OUTPUT.put_line( 'Compression Type                                                       : ' ||
l_comptype_str);
END;
/
```

Scratch Tablespace Name

Schema Name

Object Name

Compression Type

Number of Rows to Estimate

Object Type

Output of Compression Advisor Estimate for Advanced Row Compression (Entire Table)

```
Number of blocks used by the compressed sample of the object     : 165
Number of blocks used by the uncompressed sample of the object   : 629
Number of rows in a block in compressed sample of the object     : 599
Number of rows in a block in uncompressed sample of the object   : 157
Estimated Compression Ratio of Sample                            : 3.8
Compression Type                                                 : “Compress Advanced”
```

Example: Estimating Compression Ratio for Advanced Index Compression (LOW)

```
SET SERVEROUTPUT ON
DECLARE
  l_blkcnt_cmp          PLS_INTEGER;
  l_blkcnt_uncmp        PLS_INTEGER;
  l_row_cmp             PLS_INTEGER;
  l_row_uncmp           PLS_INTEGER;
  l_cmp_ratio           NUMBER;
  l_comptype_str        VARCHAR2(32767);
BEGIN
  DBMS_COMPRESSION.GET_COMPRESSION_RATIO (
          scratchtbsname => 'USERS' ,
          ownname        => 'TEST' ,
          objname        => 'SALES_IDX' ,
          subobjname     =>  NULL ,
          comptype       =>  DBMS_COMPRESSION.COMP_INDEX_ADVANCED_LOW,
          blkcnt_cmp     => l_blkcnt_cmp,
          blkcnt_uncmp   => l_blkcnt_uncmp,
          row_cmp        => l_row_cmp,
          row_uncmp      => l_row_uncmp,
          cmp_ratio      => l_cmp_ratio,
          comptype_str   => l_comptype_str,
          subset_numrows=> DBMS_COMPRESSION.comp_ratio_minrows,
          objtype        => DBMS_COMPRESSION.objtype_index
);
DBMS_OUTPUT.put_line( 'Number of blocks used by the compressed sample of the object          : ' ||
l_blkcnt_cmp);
DBMS_OUTPUT.put_line( 'Number of blocks used by the uncompressed sample of the object        : ' ||
l_blkcnt_uncmp);
DBMS_OUTPUT.put_line( 'Number of rows in a block in compressed sample of the object          : ' ||
l_row_cmp);
DBMS_OUTPUT.put_line( 'Number of rows in a block in uncompressed sample of the object        : ' ||
l_row_uncmp);
DBMS_OUTPUT.put_line ('Estimated Compression Ratio of Sample                                 : ' ||
l_cmp_ratio);
DBMS_OUTPUT.put_line( 'Compression Type                                                       : ' ||
l_comptype_str);
END;
/
```

Output of Compression Advisor Estimate for Advanced Index Compression (LOW)

```
Number of blocks used by the compressed sample of the object      : 243
Number of blocks used by the uncompressed sample of the object    : 539
Number of rows in a block in compressed sample of the object      : 499
Number of rows in a block in uncompressed sample of the object    : 145
Estimated Compression Ratio of Sample                             : 2.2
Compression Type                                                  : “Compress Advanced Low”
```

Syntax for GET_COMPRESSION_RATIO for LOBs:

```
DBMS_COMPRESSION.GET_COMPRESSION_RATIO (
 scratchtbsname          IN     VARCHAR2,
 tabowner                IN     VARCHAR2,
 tabname                 IN     VARCHAR2,
 lobname                 IN     VARCHAR2,
 partname                IN     VARCHAR2,
 comptype                IN     NUMBER,
 blkcnt_cmp              OUT    PLS_INTEGER,
 blkcnt_uncmp            OUT    PLS_INTEGER,
 lobcnt                  OUT    PLS_INTEGER,
 cmp_ratio               OUT    NUMBER,
 comptype_str            OUT    VARCHAR2,
 subset_numrows          IN     number DEFAULT COMP_RATIO_LOB_MAXROWS);
```

Example: Estimating Compression Ratio for Advanced LOB Compression (MEDIUM)

```
SET SERVEROUTPUT ON
DECLARE
 l_blkcnt_cmp          PLS_INTEGER;
 l_blkcnt_uncmp        PLS_INTEGER;
 l_row_cmp             PLS_INTEGER;
 l_lobcnt              PLS_INTEGER;
 l_cmp_ratio           NUMBER;
 l_comptype_str        VARCHAR2(32767);
BEGIN
  DBMS_COMPRESSION.GET_COMPRESSION_RATIO (
        scratchtbsname=> 'USERS' ,
        tabowner      => 'TEST' ,
        tabname       => 'PARTS' ,
        lobname       =>  'PART_DESCRIPTION' ,
        partname      =>  NULL ,
        comptype      =>  DBMS_COMPRESSION.COMP_LOB_MEDIUM,
        blkcnt_cmp    => l_blkcnt_cmp,
        blkcnt_uncmp  => l_blkcnt_uncmp,
        row_cmp       => l_row_cmp,
        lobcnt        => l_lobcnt,
        cmp_ratio     => l_cmp_ratio,
        comptype_str  => l_comptype_str,
        subset_numrows=> DBMS_COMPRESSION.comp_ratio_lob_maxrows
);
DBMS_OUTPUT.put_line( 'Number of blocks used by the compressed sample of the object          : ' ||
l_blkcnt_cmp);
DBMS_OUTPUT.put_line( 'Number of blocks used by the uncompressed sample of the object        : ' ||
l_blkcnt_uncmp);
DBMS_OUTPUT.put_line( 'Number of rows in a block in compressed sample of the object          : ' ||
l_row_cmp);
DBMS_OUTPUT.put_line( 'Number of LOBS actually sampled                                       : ' ||
l_lobcnt);
DBMS_OUTPUT.put_line ('Estimated Compression Ratio of Sample                                 : ' ||
l_cmp_ratio);
```

```
DBMS_OUTPUT.put_line( 'Compression Type                                    : ' ||
1_comptype_str);
END;
/
```

Output of Compression Advisor Estimate for Advanced LOB Compression (MEDIUM)

```
Number of blocks used by the compressed sample of the object      : 199
Number of blocks used by the uncompressed sample of the object    : 389
Number of rows in a block in compressed sample of the object      : 293
Number of LOBS actually sampled                                   : 55
Estimated Compression Ratio of Sample                             : 1.9
Compression Type                                                  : “Compress Medium”
```

Understanding Compression Advisor Results

The example advisor output below, the result of running the advisor code above for Advanced Row Compression (Compress Advanced), shows the type of output that is possible with compression advisor.

```
Number of blocks used by the compressed sample of the object      : 165
Number of blocks used by the uncompressed sample of the object    : 629
Number of rows in a block in compressed sample of the object      : 599
Number of rows in a block in uncompressed sample of the object    : 157
Estimated Compression Ratio of Sample                             : 3.8
Compression Type                                                  : “Compress Advanced”
```

In this example, the “Estimated Compression Ratio of Sample” for Advanced Row Compression (Compress Advanced) determined by compression advisor, is 3.8x.This represents an approximate space reduction of 74% the table when compressed with Advanced Row Compression.

Compression advisor typically provides accurate estimates, of the actual compression results obtained after implementing compression. In general, typical compression ratios for data, indexes and LOBS includes:

- OLTP Table Compression and Advanced Row Compression users can typically expect compression ratios in the range of 2x to 4x
- Hybrid Columnar Compression users can typically expect compression ratios in the range of 6x to 15x
- Advanced Index Compression users can typically expect compression ratios in the range of 2x to 5x
- Advanced LOB Compression users can typically expect compression ratios in the range of 2x to 3x

Note: The compression ratio achieved, in a given environment, depends on the nature of the data being compressed.

It is important to note that compression advisor builds two temporary tables (for comparison purposes) as part of the estimation process for Advanced Row Compression (Hybrid Columnar Compression uses four tables). The temporary tables are created using the prefix 'cmp3$' and/or 'cmp4$' and are dropped by the compression advisor when no longer required. Although these temporary tables are removed after

compression advisor completes, you will need available free space for compression advisor to build the temporary tables.

For additional information about using DBMS_COMPRESSION, please see the Oracle Database documentation.

Compression Advisor Best Practices

- If you get this type of message when estimating Hybrid Columnar Compression:

  ORA-12801: error signaled in parallel query server P002
  ORA-64307: Exadata Hybrid Columnar Compression is not supported for tablespaces on this storage type
  Solution: Disable parallel processing for the session (set parallel_max_servers=0)

- Compression adviser has the restriction that the scratch tablespace cannot be uniform
- In earlier releases, Oracle did require 1M rows in a table for estimating HCC compression ratios with compression advisor – this restriction was removed in Oracle Database release 12.1.0.2 and above.
- Outside compression advisor, there are no restrictions with Hybrid Columnar Compression in regards to the minimal amount of data needed (in tables/partitions) with HCC.

More Information

For more information, and examples, about compression advisor, please see this MOS note:

How Does Compression Advisor Work (DOC ID: **1284972.1)**