ORACLE

Overview of Oracle Advanced Compression

July 2025, Version 23ai
Copyright ©2025, Oracle and/or its affiliates
Public

Purpose Statement

This document provides an overview of features and enhancements included in release 23ai. It is intended solely to help you assess the business benefits of upgrading to 23ai and planning for the implementation and upgrade of the product features described.

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, timing, and pricing 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.

Copyright ©2025, Oracle and/or its affiliates / Public

ORACLE

Table of Contents

Copyright ©2025, Oracle and/or its affiliates / Public

Introduction

The amount of data that enterprises are storing, and managing, is growing rapidly - various industry estimates indicate that data volume is doubling every 2-3 years. This exponential growth of data presents daunting challenges for organizations.

First, and foremost, are storage costs. Even though the cost of storage has been declining dramatically, the enormous growth in the volume of data still makes storage one of the biggest cost elements of most IT budgets. In addition, as databases grow at accelerating rates, it is difficult to continue to meet query performance requirements while staying within budget.

Oracle Advanced Compression, and Oracle Database, together provide a robust set of compression, performance and data storage optimization capabilities that enable organizations to succeed in this complex environment. Oracle Advanced Compression provides a comprehensive set of compression capabilities to help improve query performance, while reducing storage costs. It allows organizations to significantly reduce their overall database storage footprint by enabling compression for all types of data –relational (table), unstructured (file), index, network, and backup data.

Although storage cost savings and optimization across servers (production, development, QA, test, backup etc...) are often seen as the most tangible benefits, the features of Advanced Compression are designed to provide benefits for all components of your IT infrastructure, including memory, network bandwidth and storage. Whether it is a cloud or an on-premise Oracle Database deployment, Advanced Compression can deliver robust compression across different environments with no changes in SQL or applications.

Advanced Compression provides a comprehensive set of compression features designed to reduce costs and improve query performance by enabling compression for structured data, unstructured data, indexes, database backups, network traffic and for Data Guard redo. Each of these Advanced Compression features is described in this document.

Advanced Row Compression (tables)

Advanced Row Compression maintains compression during all types of data manipulation operations, including conventional DML such as INSERT and UPDATE. In addition, Advanced Row Compression minimizes the overhead of write operations on compressed data, making it suitable for transactional/OLTP environments as well as data warehouses, extending the benefits of compression to all application workloads.

Advanced Row Compression uses a compression algorithm specifically designed to work with OLTP/DW applications. The algorithm works by eliminating duplicate values within a database block, even across multiple columns. Compressed blocks contain a structure, called a symbol table, which 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 stored in the block header. When compared with compression algorithms that maintain a global database symbol table, Oracle's approach offers significant benefits by not introducing additional IO (needed with a global symbol table) when accessing compressed data.

The compression ratio achieved 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 up to 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.

Copyright ©2025, Oracle and/or its affiliates / Public

Advanced Row Compression query performance optimizations include:

Compression Designed for Query Performance

A significant advantage of having compression built into the database is the ability of Oracle Database to read compressed blocks (data and indexes) directly, in memory, without uncompressing the blocks. This helps improve query performance due to the reduction in IO, and the reduction in system calls related to the IO operations. Further, the buffer cache becomes more efficient by storing more data without having to add memory.

Deferred Compression

While Advanced Row Compression typically has no adverse impact on read operations, there can be additional work performed while writing data, making it impossible to eliminate performance overhead for write operations. There are several optimizations that minimize this overhead for Advanced Row Compression.

A key optimization is that Oracle Database compresses blocks in batch mode rather than compressing data every 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 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 the database determines that the block can no longer benefit from further compression. Only the transaction that performs the compression of the block will experience the minimal compression related overhead – the majority of DML transactions on compressed blocks will have similar performance as they would with uncompressed blocks.

Partial Compression

With Advanced Row Compression, when the block is full, it is compressed. More rows are then added (since more rows can now fit into the block) and the process of recompression is repeated several times until the rows in the block cannot be compressed further. Blocks are usually compressed and reformatted in their entirety, but in some cases, the block can be partially compressed, hence resulting in CPU savings and extra compression.

The partial compression feature is used, transparently by the database, on already compressed blocks (i.e., compressed with Advanced Row Compression). The database looks for uncompressed rows and transforms those into a compressed form, adding or reusing symbols from the block dictionary. This is faster than recompressing the whole block again. Full block recompression also requires that no rows are locked in the block or, that all the rows in the block are locked by the transaction inserting rows into the block.

Partial compression gets around these requirements by locking and compressing only those rows that are uncompressed and unlocked. Meaning, it can take place in the presence of other uncommitted transactions in the block.

Array Inserts

During array inserts, the database estimates the number of rows that would fit into a compressed block. All these rows are buffered, compressed, and a full block image is generated. This means that typically compression occurs only once, or twice per block, as opposed to occurring potentially many times.

Copyright ©2025, Oracle and/or its affiliates / Public

The database also obtains a much better compression ratio because the database can compress many more rows together (hence being able to extract common symbols more effectively). The algorithms also adaptively vary the number of buffered rows and increases the number of rows buffered depending on running estimates of how many compressed rows would fit into a block.

With this feature, tables with Advanced Row Compression typically experience faster scans.

Advanced Index Compression (indexes)

Indexes are typically extensively used inside OLTP databases since they are capable of efficiently supporting a wide variety of access paths to the data stored in relational tables. It is very common to find large numbers of indexes created on a single table to support the multitude of access paths for OLTP applications. This can cause indexes to contribute a greater share to the overall storage of a database when compared to the size of the base tables alone.

Creating an index using Advanced Index Compression reduces the size of all supported unique and non-unique indexes -- while still providing efficient access to the indexes. With Advanced Index Compression, indexes typically experience up to a 2x to 5x compression ratio. Advanced Index Compression works well on all supported indexes, including those indexes that are not good candidates, which includes:

- Indexes with no duplicate values, or few duplicate values for given number of leading columns of the index.

Advanced Index Compression supports two levels of compression – **LOW** and **HIGH**.

- Advanced Index Compression (LOW) simplifies index key compression. When compressing an index, it automatically computes an optimal prefix count for every index leaf block in the index, rather than using a static prefix count for all index leaf block as is done with Prefix Compression
    - **LOW** compression is for both OLTP and data warehouse applications
- Advanced Index Compression (HIGH) works at the block level to provide the best compression for each block. This means that users do not need knowledge of data characteristics – Advanced Index Compression automatically chooses the right compression per block. The "HIGH" level of Advanced Index Compression can provide significant space savings
    - **HIGH** compression is for applications that are read only/mostly, such as data warehouse applications

Index-Organized Tables (IOT's) are essentially indexes and can be compressed with Advanced Index Compression LOW.

Advanced LOB Compression (SecureFiles LOBs)

Advanced LOB Compression utilizes industry standard compression algorithms to minimize the storage requirements of SecureFiles LOB segments. With Advanced LOB Compression, LOBs such as documents or XML files typically experience up to a 2x to 3x compression ratio. Advanced LOB Compression also automatically avoids compressing LOBs that would not benefit from compression, such as a document already compressed via a 3rd party tool.

There are three levels of Advanced LOB Compression: **LOW**, **MEDIUM**, and **HIGH**:

Copyright ©2025, Oracle and/or its affiliates / Public

- **LOW** is optimized for high performance. LOW maintains about 80% of the compression achieved through MEDIUM, while utilizing typically less CPU
- By default, Advanced LOB Compression uses the **MEDIUM** level, which typically provides good compression with a modest CPU overhead
- **HIGH** achieves the highest storage savings but incurs the most CPU overhead

Advanced LOB Deduplication (SecureFiles LOBs)

It is extremely common for applications to store exact replicas of files. A typical example is an email application where multiple users may receive the same attachment. Advanced LOB Deduplication eliminates duplicate copies of SecureFiles LOBs. Oracle Database stores one image and replaces the duplicate copies with references to this image.

Consider an email application where 10 users receive an email with the same 1MB attachment. Without Advanced LOB Deduplication, the system would store one copy of the file for each of the 10 users – requiring 10MB of storage. If the email application in our example uses Advanced LOB Deduplication, it will store the 1MB attachment just once. That is a 90% savings in storage requirements.

In addition to the storage savings, Advanced LOB Deduplication also benefits application performance. Specifically, write and copy operations are much more efficient since only references to the LOB are written.

Backup Compression (RMAN and Data Pump)

In addition to compressing data stored inside the database, Advanced Compression also includes the capability to compress backup data. Recovery Manager (RMAN) and Data Pump are commonly used tools to backup an Oracle Database.

RMAN makes a block-by-block backup of the database data, also known as a "physical" backup, which can be used to perform database, tablespace, or block level recovery. Data Pump is used to perform a "logical" export by offloading data from one or more tables into a flat file.

Advanced Compression includes the capability to compress the backups generated by these tools.

RMAN Backup Compression

Advanced Compression includes RMAN compression technology that can dramatically reduce the storage requirements for backup data. Due to RMAN's tight integration with Oracle Database, backup data can be compressed before being written to disk, providing an enormous reduction in storage and a potentially large reduction in backup and restore times.

There are three levels of RMAN Compression included with Advanced Compression: **LOW**, **MEDIUM**, and **HIGH**.

The three levels can be categorized as such:

- **HIGH** - Best suited for backups over slower networks where the limiting factor is network speed
- **MEDIUM** - Recommended for most environments. Good combination of compression ratios and speed
- **LOW** - Least impact on backup throughput and suited for environments where CPU resources are the limiting factor

If you are IO-limited, but have idle CPU, then HIGH could work best. HIGH uses more CPU but saves the most space and thus gives the biggest decrease in the amount of IO required to write the backup files. On the other

Copyright ©2025, Oracle and/or its affiliates / Public

hand, if you are CPU-limited, then LOW or MEDIUM probably makes more sense. Less CPU is used, and about 80% of the space savings will typically be realized (compared to Basic RMAN compression).

It is important to note that data, indexes, and LOBS, compressed using Advanced Row Compression, Advanced Index Compression and Advanced LOB Compression remain compressed during RMAN backup and recovery. This typically results in reduced backup, and recovery, times due to the smaller footprint of the backup.

Data Pump Export Compression

Data Pump compression is fully inline on the import side, so there is no need to decompress an export file before importing it. The compressed dump file sets are automatically decompressed during import without any additional steps by the Database Administrator.

Full Data Pump functionality is available using a compressed file. Any command used on a regular file will also work on a compressed file.

The following options are used to determine which parts of a dump file set should be compressed:

- **ALL -** Enables compression for the entire export operation
- **DATA-ONLY** - Results in all data being written to the dump file in compressed format
- **METADATA-ONLY** - Results in all metadata being written to the dump file in compressed format. This is the default
- **NONE** - Disables compression for the entire export operation

An expdp command-line option for Oracle Data Pump Export can be used to control the degree of compression used (BASIC, LOW, MEDIUM, or HIGH) for an Oracle Data Pump dump file – the same options can also be specified to the PL/SQL DBMS_DATAPUMP package.

The higher the degree of compression, the higher the latency incurred but the better compression ratio achieved. That is, the HIGH option will likely incur more overhead, but should compress the data better. These options enable an organization to trade off time spent compressing data against the size of the Oracle Data Pump dump file. The reduction in dump file size will vary based on data types and other factors.

When importing using Data Pump, the CREATE TABLE statements will have compression clauses that match the definition in the export file. If a compression clause is missing, then the table inherits the COMPRESSION attributes of the tablespace where the table is stored.

Advanced Network Compression (Network Traffic)

Advanced Network Compression, also referred to as SQL Network Data Compression, compresses the network data transmitted at the sending side and then uncompresses it at the receiving side to reduce the network traffic. Advanced Network Compression reduces the size of the session data unit (SDU) transmitted over a data connection. Reducing the size of data reduces the time required to transmit the SDU.

The benefits of Advanced Network Compression include:

- **Increased effective network throughput:** Compression allows transmission of large data in less time. SQL query response becomes faster due to the reduced transmission time. Constrained bandwidth environments can utilize this to reduce query response time

Copyright ©2025, Oracle and/or its affiliates / Public

- **Reduced bandwidth utilization:** Compression saves bandwidth by reducing the data transmitted, allowing other applications to use the freed-up bandwidth. This also helps in reducing the cost of providing network bandwidth

Advanced Network Compression can not only help make SQL query responses faster, but also can help save bandwidth. On narrow bandwidth connections, with faster CPU, it could significantly improve performance.

The compression is transparent to client applications.

Data Guard Redo Transport Compression (Redo)

Oracle Data Guard provides the management, monitoring, and automation software infrastructure to create, maintain and monitor one or more standby databases to protect enterprise data from failures, disasters, errors, and data corruptions.

Data Guard maintains synchronization of primary and standby databases using redo data (the information required to recover a transaction). As transactions occur in the primary database, redo data is generated and written to the local redo log files.

Data Guard Redo Transport Services transfer redo data to the standby site(s). With Advanced Compression, redo data may be transmitted in a compressed format to reduce network bandwidth consumption and in some cases reduce transmission time of redo data.

Redo data can be transmitted in a compressed format when the Oracle Data Guard configuration uses either synchronous redo transport (SYNC) or asynchronous redo transport (ASYNC).

Compression Best Practices

Below are some best practices, and considerations, regarding the features of Advanced Compression:

- The general recommendation is to compress all application related tables in the database with one exception: if the table is used as a queue. That is, if the rows are inserted into the table, then later most, or all, of the rows are deleted, then more rows are inserted and then again deleted. This type of activity is not a good use case for compression due to the overhead to constantly compress rows that are transient in nature

- The best test environment for each Advanced Compression feature is where you can most closely duplicate the production environment– this will provide the most realistic (pre- and post- compression) performance and functionality comparisons

- Space usage reduction with Advanced Row Compression gives the best results where the most duplicate data is stored (low cardinality). This is especially true for backups -- greater compression will result in less data backed up and hence shorter recovery time. Sorting data (on the columns with the most duplicates) prior to bulk loads may increase the compression ratio

- Although CPU overhead is typically minimal, implementing Advanced Row Compression is ideal on systems with available CPU cycles, as compression will have additional overhead for some DML operations

- Compression Advisor is a PL/SQL package that is used to estimate potential storage savings, for Advanced Row Compression, based on analysis of a sample of data

Copyright ©2025, Oracle and/or its affiliates / Public

It provides a good estimate of the actual compression ratio after implementing Advanced Row Compression, Advanced Index Compression or Advanced LOB Compression. Compression Advisor (DBMS_COMPRESSION) is included with Oracle Database Enterprise Edition

- Larger blocks do not always ensure higher Advanced Row Compression ratios. Testing with your own data is suggested if you want to determine if larger/smaller block sizes will have an impact on your Advanced Row Compression ratio

- It is recommended that LOBs over 4K in size be managed using SecureFiles. The Advanced LOB Compression and Deduplication features of Advanced Compression reduce the amount of storage required for SecureFiles LOB segments

- Data Pump data compression is completely independent of Advanced Row Compression. The Data Pump dump file is uncompressed inline during the import process, and the data is then imported into the target table based on the compression characteristics of the table

- Before Oracle Database 12c Release 2, blocks containing many types of chained rows could not be compressed. This limitation has been removed in Oracle Database 12c Release 2 and above

- Index-Organized Tables (IOT's) can be compressed using either Prefix Compression or Advanced Index Compression LOW

- Index Key (prefix) Compression can be very beneficial when the prefix columns of an index are repeated many times within a leaf block. However, if the leading columns are very selective, or if there are not many repeated values for the prefix columns, then Index Key Compression would not be beneficial

- Advanced Row Compression works well with Oracle Transparent Data Encryption (TDE) tablespace-level encryption. Tables are compressed before encryption, so the compression ratio is not affected by the encryption. With column-level encryption, the encryption is done before compression, which will negatively impact the compression ratio

- You cannot specify any type of table compression for external tables or for tables that are part of a cluster

- You cannot specify any type of compression for tables with LONG or LONG RAW columns, tables that are owned by the SYS schema and reside in the SYSTEM tablespace, or tables with ROWDEPENDENCIES enabled

Enabling Compression for Existing Tables

To enable compression for existing tables and indexes, those segments must be rebuilt with compression enabled. This operation can be performed online, as to allow processing to continue as compression is enabled.

Copyright ©2025, Oracle and/or its affiliates / Public

For new tables and partitions, enabling Advanced Row Compression is easy: simply CREATE the table or partition and specify "**ROW STORE COMPRESS ADVANCED**".

Example:

```
CREATE TABLE emp (emp_id NUMBER, first_name VARCHAR2(128),
last_name VARCHAR2(128)) ROW STORE COMPRESS ADVANCED;
```

For existing tables and partitions/subpartitions, there are several recommended approaches to enabling Advanced Row Compression online:

- **Online Redefinition (DBMS_REDEFINITION)**

  This approach will enable Advanced Row Compression for future DML and will compress existing data. Using DBMS_REDEFINITION keeps the table online for both read/write activity during the migration.

  Run DBMS_REDEFINITION in parallel for best performance.

- **ALTER TABLE MOVE ONLINE**

  This approach will also enable Advanced Row Compression for future DML and will compress existing data. ALTER TABLE MOVE ONLINE enables DML operations to continue to run uninterrupted on the table being moved. The ONLINE option is included with Advanced Compression.

  Run DBMS_REDEFINITION in parallel for best performance.

Please see the current Oracle Database documentation for additional details, usage examples and restrictions regarding these operations.

Conclusion

The massive growth in data volume, experienced by enterprises, introduces significant challenges. Companies must quickly adapt to the changing business landscape without influencing the bottom line. Organizations need to efficiently manage their existing infrastructure to control costs while continuing to deliver extraordinary application query performance.

Advanced Compression, along with Oracle Database, together provide a robust set of compression, query performance and data storage optimization capabilities that enable organizations to succeed in this complex environment.

Using Advanced Compression, organizations can efficiently manage their increasing data requirements throughout all components of their data center – minimizing storage-related costs while continuing to achieve the highest levels of application query performance.

Get Started with Compression Advisor

An easy way to get started, with Advanced Compression, is by using 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 or partition that was the target of compression advisor. The output indicates the "COMPRESSION RATIO"

Copyright ©2025, Oracle and/or its affiliates / Public

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.

Compression Advisor is included with Oracle Database Enterprise Edition.

More Information

- See the Oracle *Database Administrator's Guide* documentation for more information about Oracle compression.
- See the Oracle *VLDB and Partitioning Guide* documentation for more information about Heat Map and Automatic Data Optimization
- See the *PL/SQL Packages and Types Reference* for more information, and usage examples, about the DBMS_SECUREFILES.GET_LOB_DEDUPLICATION_RATIO function, Compression Advisor (DBMS_COMPRESSION) and Online Redefinition (DBMS_REDEFINITION)

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com facebook.com/oracle twitter.com/oracle

Copyright © 2025, Oracle and/or its affiliates. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.