

ORACLE

Data, Index, LOB, Backup and Network Compression with Oracle Advanced Compression

January, 2021, Version 21.2
Copyright © 2021, Oracle and/or its affiliates
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

Purpose statement	2
Disclaimer	2
Introduction	4
Data Compression	4
Compression Designed for Performance	5
Deferred Compression	5
Partial Compression	6
Array Inserts	6
Index Compression	6
LOB Compression	7
LOB Deduplication	7
Backup Compression	8
RMAN/Export Backup Compression	8
DataPump Export Compression	8
Network Compression	9
Data Guard Redo Transport Compression	10
Compression Best Practices	10
Enabling Compression for Existing Tables	11
Conclusion	12
Free Compression Advisor	13

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 IT. 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 performance requirements while staying within budget.

Oracle is a pioneer in database compression technology. Oracle Advanced Compression, and Oracle Database, together provide a robust set of compression, performance and data storage optimization capabilities that enable IT managers to succeed in this complex environment. Oracle Advanced Compression provides a comprehensive set of compression capabilities to help customers improve performance while reducing storage costs. It allows IT administrators 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, all of the features of Advanced Compression are designed to improve performance for all components of your IT infrastructure, including memory, network bandwidth and storage. Whether it is a cloud or an on-premises Oracle database deployment, Oracle Advanced Compression can deliver robust compression across different environments with no changes in applications. Benefits from Oracle Advanced Compression include smaller database storage footprint, savings in backups and improved database performance.

Oracle Advanced Compression provides a comprehensive set of compression features designed to reduce costs and improve 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.

Data Compression

Advanced Compression, with Oracle Database 11g Release 1, introduced OLTP Table Compression, now called Advanced Row Compression, which 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 unique compression algorithm specifically designed to work with OLTP/DW applications. The algorithm works by eliminating duplicate values within a database block, even across multiple

Defer Storage Purchases

Advanced Row Compression is a non-obtrusive data compression solution that typically results in a 50%, or more, reduction in overall storage requirement for database tables with little or no disruption for implementation. This will allow you to better utilize existing database storage, and possibly defer additional purchases of new storage, for growing applications. Because Advanced Compression substantially reduces the total data size in the application database, storage requirements grow much slower than non-compressed databases.

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 competing compression algorithms that maintain a global database symbol table, Oracle's approach offers significant performance benefits by not introducing additional I/O (needed with a global symbol table) when accessing compressed data.

Note that 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.

Compression Designed for 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 performance due to the reduction in I/O, and the reduction in system calls related to the I/O operations. Further, the buffer cache becomes more efficient by storing more data without having to add memory.

Additional Advanced Compression performance optimizations include:

Deferred Compression

As described above, Advanced Row Compression has no adverse impact on read operations. Although there can be additional work performed while writing data, making it impossible to completely 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 Oracle determines that the block can no longer benefit from further compression. Only the transaction that performs the compression of the block will experience the slight compression overhead – the majority of DML transactions on compressed blocks will have the exact same 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 Oracle 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 - hence it can take place in the presence of other uncommitted transactions in the block.

Array Inserts

During array inserts, Oracle 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 (one compression to estimate compression ratio) per block, as opposed to occurring potentially many times. This optimization provides a significant benefit in elapsed time. Oracle Database also obtains a much better compression ratio because Oracle 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 not only enable much faster scans, but also faster inserts than uncompressed tables.

Index Compression

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 a large number 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.

Advanced Index Compression, a feature of Advanced Compression, is a form of index block compression. 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. Advanced Index Compression works well on all supported indexes, including those indexes that are not good candidates (indexes with no duplicate values, or few duplicate values for given

number of leading columns of the index) with the existing Prefix Compression feature.

Advanced Index Compression 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 provides significant space savings while also improving performance for queries that are executed using indexes.

LOB Compression

Advanced Compression provides a mechanism to control the size of your SecureFiles managed data. Advanced LOB Compression, a feature of Advanced Compression, utilizes industry standard compression algorithms to minimize the storage requirements of SecureFiles LOBs. With Advanced LOB Compression, files such as documents or XML files typically experience a 2x to 3x compression ratio. Advanced LOB Compression automatically avoids compressing data that would not benefit from compression – for instance, a document compressed via a 3rd party tool before being inserted into the database as a SecureFiles LOB.

Applications are still able to perform random reads and writes on compressed SecureFiles data since the compressed data is internally broken down into small chunks of data. This can vastly improve performance when compared with compressing entire files before inserting them into the database.

There are three levels of Advanced LOB Compression: LOW, MEDIUM, and HIGH. By default, Advanced LOB Compression uses the MEDIUM level, which typically provides good compression with a modest CPU overhead of 3%-5%. Advanced LOB Compression LOW is optimized for high performance. Advanced LOB Compression LOW maintains about 80% of the compression achieved through MEDIUM, while utilizing typically 3x less CPU. Finally, Advanced LOB Compression HIGH achieves the highest storage savings but incurs the most CPU overhead.

LOB Deduplication

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, a feature of Advanced Compression, eliminates duplicate copies of SecureFiles LOBs. Oracle Database stores one image of the SecureFiles data 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 increases application performance. Specifically, write and copy operations are much more efficient since only references to the SecureFiles data are written. Further, read

operations may improve if duplicate SecureFiles data already exists in the buffer cache.

Backup Compression

In addition to compressing data stored inside the database, Advanced Compression also includes the capability to compress backed up data. Recovery Manager (RMAN) and Data Pump are the two most commonly used tools to backup the data stored inside 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” backup by offloading data from one or more tables into a flat file.

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

RMAN Backup Compression

The continuous growth in enterprise databases creates an enormous challenge to database administrators. The storage requirements for maintaining database backups, and the performance of the backup procedures, are directly impacted by database size. 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 costs 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. Generally speaking, 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

As indicated above, if you are I/O-limited but have idle CPU, then HIGH could work best, as it uses more CPU, but saves the most space and thus gives the biggest decrease in the number of I/O's required to write the backup files. On the other 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).

Data Pump Export Compression

Data Pump export compression is an inline operation, so the reduced dump file size typically means a significant savings in disk space. Unlike operating system or file system compression utilities, Data Pump compression is fully inline on the import side as well, so there is no need to decompress a dump 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 the DBA 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.

Note that 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.

Network Compression

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
- **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 not only makes SQL query responses faster but also saves 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

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 of the 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, although minor 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. It provides a good estimate of the actual compression ratio after implementing Advanced Row Compression. A version of Compression Advisor, which supports Oracle Database 9i Release 2 through 11g Release 1, is available free on the Advanced Compression page on Oracle.com. Compression Advisor

(DBMS_COMPRESSION) is included with Oracle Database 11g Release 2 and above

- Advanced Row Compression is NOT supported for use with tables that have LONG data types.
- 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 LOBs
- Data Pump data compression is completely independent of Advanced Row Compression. The Data Pump dumpfile 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 starting with Oracle Database 12c Release 2
- Index-Organized Tables (IOT's) are essentially indexes, so they cannot be compressed with Advanced Row or Basic Table Compression. IOT's can be compressed with Prefix Compression
- 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 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

Enabling Compression for Existing Tables

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

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

For existing tables and partitions, there are a number of recommended approaches to enabling Advanced Row Compression:

ALTER TABLE ... ROW STORE COMPRESS ADVANCED

This approach will enable Advanced Row Compression for all future DML -- however, the existing data in the table will remain uncompressed.

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.

Online redefinition will clone the indexes to the interim table during the operation. All the cloned indexes are incrementally maintained during the sync (refresh) operation so there is no interruption in the use of the indexes during, or after, the online redefinition. The only exception is when online redefinition is used for redefining a partition -- any global indexes are invalidated and need to be rebuilt after the online redefinition.

ALTER TABLE ... MOVE ROW STORE COMPRESS ADVANCED

This approach will enable Advanced Row Compression for future DML and will compress existing data. The table being moved is online for read activity, but has an exclusive (X) lock, so all DML will be blocked until the move command completes. Run ALTER TABLE...MOVE in parallel for best performance.

ALTER TABLE... MOVE will invalidate any indexes on the partition or table; those indexes will need rebuilt after the ALTER TABLE... MOVE. For partition moves, the use of ALTER TABLE... MOVE PARTITION with the UPDATE INDEXES clause will maintain indexes (it places an exclusive (X) lock so all DML will be blocked until the move command completes) – not available for non-partitioned tables.

The ALTER TABLE... MOVE statement allows you to relocate data of a non-partitioned table, or of a partition of a partitioned table, into a new segment, and optionally into a different tablespace. ALTER TABLE...MOVE ROW STORE COMPRESS ADVANCED compresses the data by creating new extents for the compressed data in the tablespace being moved to -- it is important to note that the positioning of the new segment can be anywhere within the data file, not necessarily at the tail of the file or head of the file. When the original segment is released, depending on the location of the extents, it may or may not be possible to shrink the data file.

ALTER TABLE ... MOVE TABLE/PARTITION/SUBPARTITION ... ONLINE ROW STORE COMPRESS ADVANCED

This approach will enable Advanced Row Compression for future DML and will compress existing data. ALTER TABLE ... MOVE TABLE, PARTITION or SUBPARTITION ... ONLINE enables DML operations to continue to run uninterrupted on the table/partition/subpartition being moved. Indexes are maintained during the move operation, so a manual index rebuild is not required.

Note: 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. IT managers need to efficiently manage their existing infrastructure to control costs, yet continue to deliver extraordinary application performance.

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

Using Advanced Compression, enterprises 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 performance.

Free 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” 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.

A version of Compression Advisor, which supports Oracle Database 9i Release 2 through 11g Release 1, is available free on the Advanced Compression page on Oracle.com. For later releases, a version of Compression Advisor is included with Oracle Database Enterprise Edition 11g Release 2 and above.

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 © 2021, Oracle and/or its affiliates. All rights reserved. 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.

This device has not been authorized as required by the rules of the Federal Communications Commission. This device is not, and may not be, offered for sale or lease, or sold or leased, until authorization is obtained.

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

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0120

Disclaimer: If you are unsure whether your data sheet needs a disclaimer, read the revenue recognition policy. If you have further questions about your content and the disclaimer requirements, e-mail REVREC_US@oracle.com.
