ORACLE
EXADATA

Hybrid Columnar Compression (HCC) on Oracle Database 18c

ORACLE WHITE PAPER | OCTOBER 2018

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. 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 for Oracle's products remains at the sole discretion of Oracle.

Table of Contents

Introduction

Hybrid Columnar Compression enables the highest levels of data compression and provides enterprises with tremendous cost-savings, and performance improvements, due to reduced I/O. HCC is optimized to use both database and storage capabilities on Exadata to deliver tremendous space savings and revolutionary performance. Average data compression ratios can range from 10x to 15x depending on which Hybrid Columnar Compression level is implemented – real world customer benchmarks have resulted in data compression ratios of up to 204x.

With average data compression ratios of 10x from HCC, IT managers can drastically reduce, and often eliminate, their need to purchase new storage for several years. For example, a 100 terabyte database achieving a 10x data compression ratio would utilize only 10 terabytes of physical storage. With 90 terabytes of storage now available, IT organizations could potentially delay storage purchases for a significant amount of time. The 90 terabytes of storage could even be used to store up to 9 more databases with 100 terabytes of data compressed to 10 terabytes of actual disk space each.

Hybrid Columnar Compression is an enabling technology for both HCC Warehouse (Query) Compression and HCC Archive Compression. We will discuss each of these capabilities in detail later in this document, but first we will explore the implementation, and significant benefits, of Hybrid Columnar Compression on Exadata.

Hybrid Columnar Compression

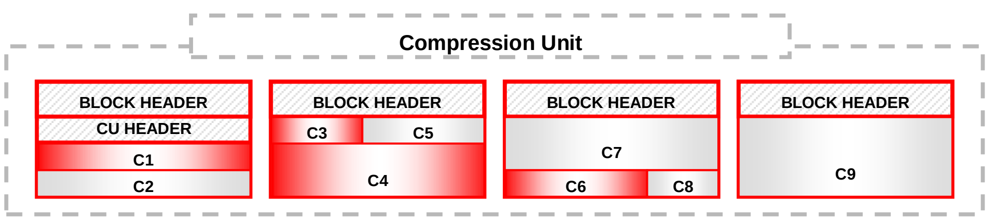
Traditionally, data has been organized within a database block in a 'row' format, where all column data for a particular row is stored sequentially within a single database block. Having data from columns with different data types stored close together limits the amount of storage savings achievable with compression technology. An alternative approach is to store data in a 'columnar' format, where data is organized and stored by column.

Storing column data together, with the same data type and similar characteristics, dramatically increases the storage savings achieved from compression. However, storing data in this manner can negatively impact database performance when application queries access more than one or two columns, perform even a modest number of updates, or insert small numbers of rows per transaction.

Oracle's Hybrid Columnar Compression technology is a new method for organizing data within a database block. As the name implies, this technology utilizes a combination of both row and columnar methods for storing data. This hybrid approach achieves the compression benefits of columnar storage, while avoiding the performance shortfalls of a pure columnar format.

A logical construct called the compression unit (CU) stores a set of hybrid columnar compressed rows. When data is loaded, column values for a set of rows are grouped together and compressed (columns are not re-ordered or combined within the compression unit). After the column data for a set of rows is compressed, it is stored in a compression unit.

Figure 1: Conceptual Illustration of a Logical Compression Unit (CU)



To maximize storage savings with Hybrid Columnar Compression, data must be loaded using data warehouse bulk loading (direct path) techniques. Examples of bulk load operations commonly used in data warehouse environments are:

- Insert statements with the APPEND hint
- Parallel DML
- Direct Path SQL*LDR
- Create Table as Select (CTAS)

Queries on hybrid columnar compressed data often run in the Exadata storage cells with Smart Scans, using a high performance query engine that utilizes special columnar processing techniques. Data sent back to the database server(s) is usually compressed (and is typically much less data than is read from disk) and the compressed data is subsequently processed by the database server(s). Note that data remains compressed not only on disk, but also remains compressed in the Exadata Smart Flash Cache, on Infiniband, in the database server buffer cache, as well as when doing RMAN back-ups or log shipping to Data Guard.

One of the key benefits of the hybrid columnar approach is that it provides both the compression, and performance, benefits of columnar storage without sacrificing the robust feature set of the Oracle Database. For example, while optimized for scan-level access, Oracle is still able to provide efficient row-level access, with entire rows typically retrieved with a single I/O, because row data is self-contained within compression units.

In contrast, pure columnar formats require at least one I/O per column for row-level access. With data warehousing tables generally having hundreds of columns, it is easy to see the performance benefits of Hybrid Columnar Compression on

Exadata. Further, tables using Hybrid Columnar Compression on Exadata still benefit from all of the high availability, performance, and security features of the Oracle Database.

While HCC compressed data can be modified using conventional Data Manipulation Language (DML) operations, such as UPDATE and DELETE - HCC is best suited for applications with no, or very limited DML operations. If frequent UPDATE and DELETE operations are planned on a table or partition, then Advanced Row Compression (a feature of Oracle Advanced Compression) is better suited for such data. Starting with Oracle Database 12.2, HCC automatically compresses new data from SQL INSERT ... SELECT statements, without the APPEND hint and array inserts from programmatic interfaces such as PL/SQL and the Oracle Call Interface (OCI).

HCC Warehouse (Query) Compression

Data warehouses have become increasingly important in the day-to-day operations of enterprises. They are responsible for storing significant amounts of data, transforming that data into strategic information, and providing management with the necessary intelligence to run the enterprise. As the importance of data warehouses has increased, so too has the amount of data managed by data warehouses. With data volumes often doubling every two years, IT Managers are experiencing significant challenges in both storage costs and application query performance. HCC Warehouse Compression is the next-generation HCC compression feature dedicated to solving both of these challenges.

HCC Warehouse (also called HCC Query) Compression provides significant storage savings by leveraging Hybrid Columnar Compression technology. Query Compression typically provides a 10:1 (10x) data compression ratio, delivering roughly five times the industry average savings. For example, enabling Query Compression on an uncompressed 100 terabyte data warehouse would reduce the storage requirements to only 10 terabytes. Query Compression would return 90 terabytes of storage back to the enterprise for other uses. In fact, the enterprise could use this reclaimed storage to support the growth of its data warehouse without purchasing additional storage for over 4 years, assuming the database doubled in size every two years. Clearly, storage savings of this magnitude dramatically reduce costs as enterprises can significantly delay storage purchases for many years.

Many data warehouse applications are hitting a performance bottleneck due to growth in data volumes. Analytical queries are scanning hundreds of gigabytes, if not terabytes, of data making the storage system the limiting factor to performance and scalability. With Oracle Exadata Storage Server, traditional and costly remedies such as purchasing additional disks to improve scan performance and I/O throughput are no longer necessary. With Exadata's massively parallel storage grid, Smart Scan capabilities and Query Compression, IT administrators are no longer forced to increase the number of disk drives in their storage arrays simply to increase performance.

While Query Compression is a database storage optimization feature, the implementation of Hybrid Columnar Compression on Exadata is optimized to improve I/O scan performance during typical data warehouse queries. The I/O required to scan a Query-compressed table typically decreases by the compression ratio achieved. Therefore, scan-oriented queries that access a table that has a compression ratio of 10:1 will likely have a reduction in I/O of up to 10x. Total query performance will also likely improve, however it will depend on the available CPU resources.

Further improving performance is Exadata's Smart Scan technology, which greatly reduces the amount of data sent from storage to the database server by offloading much of the scan activities to the Exadata storage. Exadata Smart Scan works directly on Hybrid Columnar compressed data on Exadata. With this level of improvement in I/O scan performance, Query Compression reduces costs by both decreasing the amount of storage required and by eliminating the need to increase the number of disk drives and related hardware to meet performance objectives.

Query Compression provides two levels of compression: LOW and HIGH. Query Compression HIGH typically provides a 10x reduction in storage, while Query Compression LOW typically provides a 6x reduction. Both levels are optimized on Exadata to increase scan query performance by taking advantage of the fewer number of blocks on disk.

To maximize the storage savings and query performance benefits of Query Compression, the default level is HIGH. The increased storage savings may cause data load times to increase modestly. Therefore, Query Compression LOW should be chosen for environments where load time service levels are more critical than query performance.

HCC Archive Compression

One of the biggest challenges facing IT administrators today is the cost and complexity of managing historical data. IT managers are being forced to reduce costs, yet conflicting business requirements dictate that data be kept available for significantly longer periods of time, often indefinitely.

Organizations have developed Information Lifecycle Management (ILM) strategies to help mitigate the costs of storing this data. As data ages, the typical ILM strategy involves moving data to less expensive storage, including less expensive disk drives and often archiving this data to tape. As a result, the more expensive and higher performing disk drives are used exclusively for the most recent and thus most accessed data. Hybrid Columnar Compression's Archive Compression is a new approach to reducing the storage requirements and costs of storing this historical data.

HCC Archive Compression provides significant storage savings by leveraging Hybrid Columnar Compression technology. Archive Compression is optimized to maximize storage savings, typically achieving a compression ratio of 15:1 (15x). That is, an uncompressed table or partition would require 15x more storage than a table or partition using Archive Compression.

In contrast to Query Compression, Archive Compression is a pure storage saving technology. Tables or partitions utilizing Archive Compression will typically experience a decrease in performance - a factor of the compression algorithm being optimized for maximum storage savings. Therefore, Archive Compression is intended for tables or partitions that store data that is rarely accessed.

Databases supporting any application workload, including OLTP and data warehouses, can use Archive Compression to reduce the storage requirements of historical data. Oracle supports enabling any type of table compression at the partition or sub-partition level. An OLTP application, therefore, can store historical data in partitions with Archive Compression, while active data remains in partitions with Oracle's Advanced Row Compression. Advanced Row Compression, a feature of Oracle Advanced Compression, is a compression technology that is optimized for active transactional databases. Advanced Row Compression typically provides data compression ratios of 2x - 4x, delivering significant savings to OLTP databases.

Data Warehouses on Exadata will typically store frequently queried data in partitions with Query Compression (for performance), while historical data is stored in partitions with Archive Compression (for storage savings).

In many applications, historical data is responsible for consuming up to 80% of the allocated storage. It is no wonder that IT administrators are implementing ILM strategies that archive much of this historical data to tape. However, this approach has several inherent flaws. Once data is archived to tape, the application can no longer access this data directly. In order to access the archived data, IT administrators must first restore the data from tape and load it back into the database. This can take a tremendous amount of time and doesn't meet the requirements of today's fast-paced businesses.

Complicating things further, data archived to tape becomes out-of-sync with structural changes to the database schema, such as the addition of columns and constraints. Therefore, restoring this data back into the database requires not only a significant amount of time but also a significant amount of resources to correctly restore the data and make it accessible by the application. Of course, requests for this data are usually extremely urgent and any delays affect management's ability to make critical business decisions. As you can see, this approach to storing historical data can actually be quite costly to the business.

Archive Compression provides the storage savings benefits of archiving data to tape while keeping this data online for immediate access and modification. Further, as the application evolves, all the historical data will evolve with the database schema modifications, such as new columns, constraints, etc. Therefore, when an application user needs to access historical data, the application will be able to seamlessly service queries without any need to involve IT administrators or application developers.

While query performance against tables or partitions with Archive Compression is slower than tables or partitions with Query Compression or Advanced Row Compression, they are orders of magnitude faster than queries against data that is archived to tape.

Hybrid Columnar Compression Row Level Locking

Hybrid Columnar Compression uses one lock per CU. Optionally; users can choose to enable Row Level Locking for Compression Units. The default, with HCC, is NO ROW LEVEL LOCKING; ROW LEVEL LOCKING is explicitly specified during a CREATE TABLE or ALTER TABLE MOVE operation.

The following is an example of enabling HCC Row Level Locking:

… COLUMN STORE COMPRESS FOR QUERY HIGH ROW LEVEL LOCKING;

Migration and Best Practices

For new tables and partitions, enabling Hybrid Columnar Compression is as easy as simply CREATEing the table or partition and specifying a compression level, see example below:

CREATE TABLE … **COLUMN STORE COMPRESS FOR QUERY HIGH;**

The **….COMPRESS FOR QUERY HIGH** compression level is used as an example in this document, the additional compression levels, available with Hybrid Columnar Compression, include:

… COLUMN STORE COMPRESS FOR QUERY LOW

… COLUMN STORE COMPRESS FOR ARCHIVE LOW

… COLUMN STORE COMPRESS FOR ARCHIVE HIGH

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

1. **ALTER TABLE … COLUMN STORE COMPRESS FOR QUERY HIGH**
    - This approach will enable Hybrid Columnar Compression for all future DML -- however, the existing data in the table will remain uncompressed.
2. **Online Redefinition (DBMS_REDEFINITION)**
    - This approach will enable Hybrid Columnar Compression for future DML and also 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.
3. **ALTER TABLE … MOVE COLUMN STORE COMPRESS FOR QUERY HIGH**
    - This approach will enable Hybrid Columnar Compression for future DML and also compress existing data. While the table is being moved it 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 to be 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 COLUMN STORE COMPRESS FOR QUERY HIGH 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.

4. ALTER TABLE … MOVE TABLE/PARTITION/SUBPARTITION … ONLINE

This approach will enable Hybrid Columnar Compression for future DML and also compress existing data. ALTER TABLE ... MOVE PARTITION/SUBPARTITION … **ONLINE** allows DML operations to continue to run uninterrupted on the table/partition/subpartition that is being moved. Indexes are maintained during the move partition operation, so a manual index rebuild is not required. New in 12.2, move tables online as well as partitions/subpartitions.

Below are some best practices and considerations when using Hybrid Columnar Compression:

- The best test environment for Hybrid Columnar Compression is where you can most closely duplicate the production environment– this will provide the most realistic (pre- and post- compression) performance comparisons.
- Oracle Compression Advisor (DBMS_COMPRESSION) is a PL/SQL package, included with Oracle Database 11g Release 2 and above, that is used to estimate potential storage savings for Hybrid Columnar Compression based on analysis of a sample of data. It provides a good estimate of the actual results that may be obtained after implementing Hybrid Columnar Compression. It also provides compression ratio estimates for table compression using Advanced Row Compression. Compression Advisor for HCC will run on any platform that supports Oracle Database Enterprise Edition 11g Release 2 and above.
- The use of Hybrid Columnar Compression is also available for FS Flash Storage System, ZFS Storage Appliance (ZFSSA) and Oracle Database Appliance (ODA). This enables Oracle Database users to utilize Oracle's Hybrid Columnar Compression on FS Flash Storage Systems, Sun ZFS Storage Appliance (ZFSSA) storage and Oracle Database Appliance (ODA). This provides the storage benefits of Oracle's Hybrid Columnar Compression, which had previously been exclusive to Exadata storage, to Oracle Database users who use FS Flash Storage, Sun ZFSSA storage and/or Oracle Database Appliance, enabling data compression ratios of 6x to 15x, depending on the data and the compression level chosen by the user.
- Hybrid Columnar Compression is NOT supported for use with the LONG data types and the use of UNIFORM EXTENTS is not recommended with Hybrid Columnar Compression
- There are no restrictions with Hybrid Columnar Compression in regards to the minimal amount of data needed with HCC. HCC can be very effective even with only a few MB’s of data per segment/partition. However, when using smaller amounts of data (MB’s per segment) and Parallel Loads it is important to note that Parallel Loads sometime use temp segment merge, where each loader process creates a separate segment, in this scenario Oracle recommends having a couple of hundred MB’s per segment/partition.

- Hybrid Columnar Compression is designed for relational data, not for unstructured data in BLOBs (or CLOBs). LOBs are best stored in the Oracle Database as SecureFiles LOBs. Advanced LOB Compression and Advanced LOB Deduplication, features of Oracle Advanced Compression, can be used to reduce the amount of storage required for SecureFiles LOBs.
- Hybrid Columnar Compression does not compress indexes or Index-Organized Tables (IOT's). Indexes (and IOT's) can be compressed using Prefix Compression, which is included with Oracle Database Enterprise Edition.
- DML UPDATE operations, against a Hybrid Columnar Compressed table/partition, can reduce the overall compression savings over time since data UPDATED, via DML operations, will not be compressed to the same data compression ratio as other HCC compressed data.

Conclusion

With exploding data volume growth in both data warehouses and OLTP applications, IT Managers need tools to efficiently manage their IT infrastructure while controlling costs and maintaining or improving performance.

Hybrid Columnar Compression on Exadata provides the IT Manager with those exact tools - a robust set of compression features that drastically reduce infrastructure costs while improving application performance.

For more information about Hybrid Columnar Compression, please see the Storage Optimization blog here: https://blogs.oracle.com/DBStorage/

ORACLE

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

AUTHOR: GREGG CHRISTMAN

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

Hardware and Software, Engineered to Work Together