Oracle Advanced Compression

An Oracle White Paper
April 2008

Oracle Advanced Compression

INTRODUCTION

Enterprises are experiencing an explosion in the volume of data required to effectively run their businesses. This trend in data growth can be attributed to several key factors. Recent changes in the regulatory landscape, such as Sarbanes-Oxley and HIPPA, are contributing to this trend by mandating that enterprises retain vast amounts of information for long periods of time. Mass distribution of rich and multimedia content over the Internet, made possible through advancements in broadband technologies, also contribute to the growth in overall data volume. Further fueling the exponential trend in data growth is the advent of Web 2.0, where collaborative applications promote enormous amounts of user-generated content. Various estimates indicate data volume almost doubling every 2-3 years.

This sudden explosion in data volume presents a daunting management challenge for IT administrators. First and foremost is the spiraling storage costs. Even though the cost per MB of storage has been declining dramatically in the last few years, the enormous growth in the volume of data that needs to be retained online makes storage one of the biggest cost elements of the IT budget. In addition, application scalability and performance must continue to meet the demands of the business – even as data volumes explode.

Oracle Database 11g introduces the Advanced Compression Option to help customers cope with these challenges. Innovations in Oracle compression technologies help customers reduce the resources and costs of managing large data volumes. The introduction of these exciting new technologies come at an opportune time as terabyte-sized databases, once considered a novelty, are becoming prevalent in enterprise data centers.

ORACLE ADVANCED COMPRESSION

The Oracle Database 11g Advanced Compression Option introduces a comprehensive set of compression capabilities to help customers maximize resource utilization and reduce costs. It allows IT administrators to significantly reduce their overall database storage footprint by enabling compression for all types of data – be it regular relational (structured), unstructured (documents, spreadsheets, etc) or backup data. Also, although the storage cost saving is often seen as the most tangible benefit of compression, innovative technologies included in the Advanced Compression Option are designed to reduce resource requirements and technology costs for all components of your IT infrastructure, including memory and network bandwidth.

Compression for Relational Data

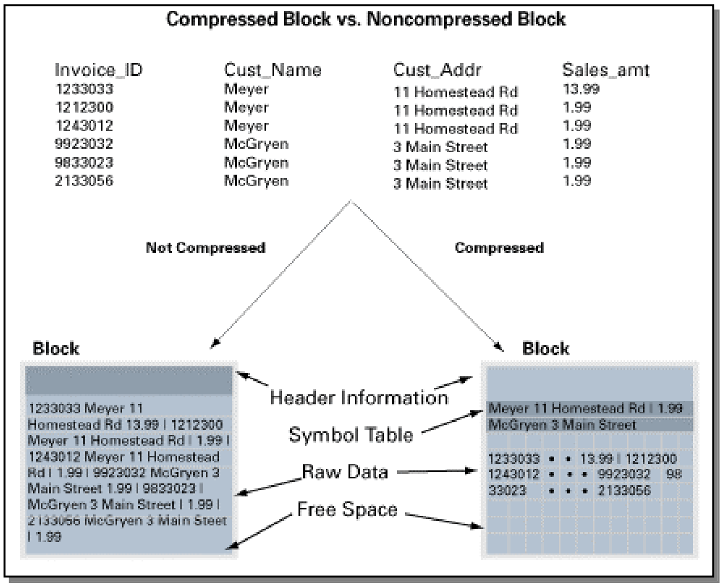
Oracle has been one of the pioneers in introducing database compression technology. Oracle Database 9i introduced Table Compression several years ago to allow data to be compressed during bulk load operations such as direct path load, CREATE TABLE AS SELECT…. (CTAS), etc.. This form of compression was ideally suited for data warehousing environments where most data is loaded in the database using batch processes. Oracle Database 11g introduces a new feature called OLTP Table Compression that allows data to be compressed during all types of data manipulation operations, including conventional DML such as INSERT and UPDATE. In addition, the new feature significantly improves performance by reducing the overhead of write operations making it suitable for transactional or OLTP environments as well. This landmark innovation, therefore, extends the benefits of compression to all application workloads.

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

Innovative Algorithm

Oracle uses a unique compression algorithm specifically designed to work with relational data. The algorithm works by eliminating duplicate values within a database block, even across multiple columns. Compressed blocks contain a structure called a symbol table that maintains compression metadata. When a block is compressed, duplicate values are eliminated by first adding a single copy of the duplicate value to the symbol table. Each duplicate value is then replaced by a short reference to the appropriate entry in the symbol table. Through this innovative design, compressed data is self-contained within the database block as the metadata used to translate compressed data into its original state is contained within the block. When compared with competing compression algorithms that maintain a global database symbol table, Oracle's unique approach offers significant performance benefits by not introducing additional I/O when accessing compressed data.

Figure 1. Compressed Block vs. Noncompressed Block



Compressed Block vs. Noncompressed Block

Invoice_ID	Cust_Name	Cust_Addr	Sales_amt
1233033	Meyer	11 Homestead Rd	13.99
1212300	Meyer	11 Homestead Rd	1.99
1243012	Meyer	11 Homestead Rd	1.99
9923032	McGryen	3 Main Street	1.99
9833023	McGryen	3 Main Street	1.99
2133056	McGryen	3 Main Street	1.99

Not Compressed
Compressed
Block
Block
1233033 Meyer 11 Homestead Rd 13.99 | 1212300 Meyer 11 Homestead Rd | 1.99 | 1243012 Meyer 11 Homestead Rd | 1.99 | 9923032 McGryen 3 Main Street 1.99 | 9833023 | McGryen 3 Main Street | 1.99 | 2133056 McGryen 3 Main Steet | 1.99
Header Information
Symbol Table
Raw Data
Free Space
Meyer 11 Homestead Rd | 1.99
McGryen 3 Main Street
1233033 • • 13.99 | 1212300
1243012 • • • 9923032 98
33023 • • • 2133056

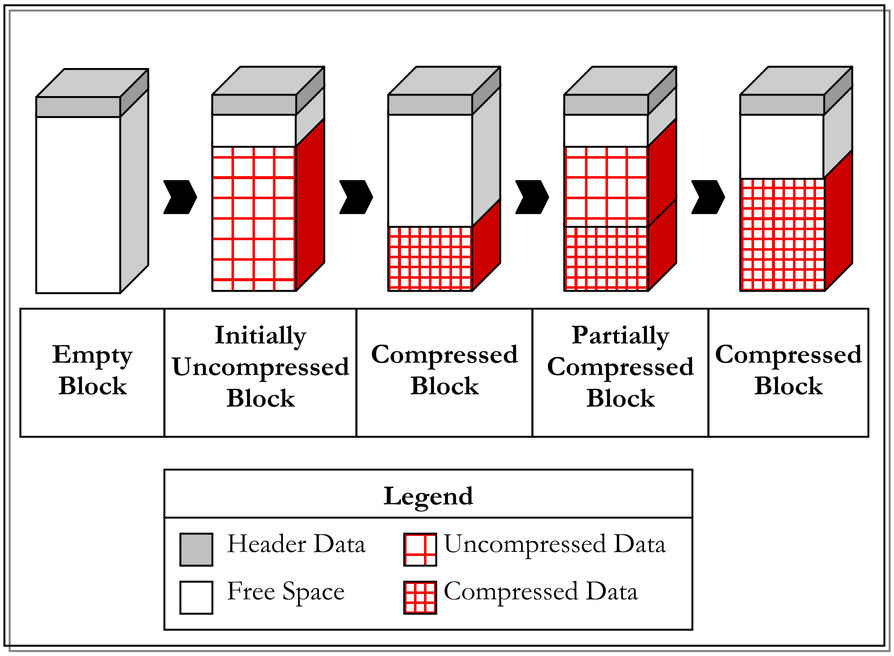
Benefits of Table Compression

The compression ratio achieved in a given environment depends on the nature of data being compressed, specifically the cardinality of the data. In general, customers can expect to reduce their storage space consumption by 2–4 times by using the table compression feature. That is, the amount of space consumed by uncompressed data volume will be two to four times larger than that of the compressed data. The benefits of compression go beyond just on-disk storage savings. One significant advantage is Oracle's ability to read compressed blocks directly without having to first uncompress the block. Therefore, there is no measurable performance degradation for accessing compressed data. In fact, in many cases performance may improve due to the reduction in I/O since Oracle will have to access fewer blocks. Further, you can achieve a more efficient buffer cache by storing more data in the cache without having to increase your memory footprint.

Minimal Performance Overhead

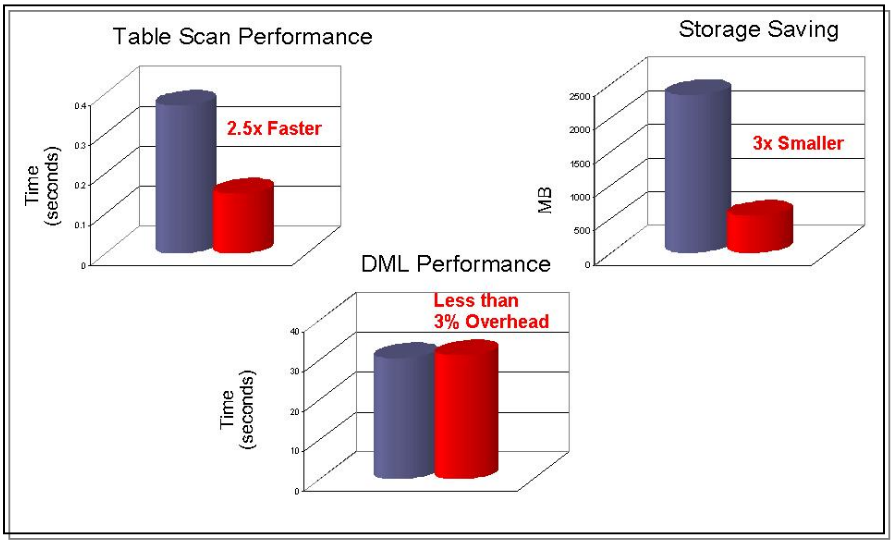
As stated above, the Table Compression feature has no adverse impact on read operations. However, compression requires additional work to be performed while writing the data making it unavoidable to eliminate performance overhead for write operations. Oracle has put in a significant amount of work to minimize such overhead for OLTP Table Compression. Oracle compresses a block in batch mode rather than compressing data every single time a write operation takes place. A newly initialized block remains uncompressed until data in the block reaches an internally controlled threshold. When a transaction causes the data in the block to reach this threshold, all contents of the block are compressed. Subsequently, as more data is added to the block and the threshold is again reached, the entire block is recompressed to achieve the highest level of compression. This process repeats until Oracle determines that the block can no longer benefit from further compression. Only transactions that trigger the compression of the block will experience the minimal compression overhead. Therefore, a majority of OLTP transactions on compressed blocks will have the exact same performance as they would with uncompressed blocks.

Figure 2. Block Compression Process



The following charts depict the performance impact of OLTP Table Compression on a real world application. These tests were conducted on the ten largest tables of an ERP database:

Figure 3. Performance Impact of Compression



As evident from these charts, Advanced Compression provides not only significant storage savings, it also provides performance improvements for read intensive workloads. For DML operations, it has a very low overhead due to the unique batch compression algorithm described above.

How to Enable Table Compression

The example below shows how OLTP Table Compression can be enabled:

```
CREATE TABLE emp (
      emp_id NUMBER
, first_name VARCHAR2(128)
, last_name VARCHAR2(128)
) COMPRESS FOR ALL OPERATIONS;
```

Similarly, Table Compression can be enabled in data warehousing environments as shown below. This is the default compression syntax:

```
CREATE TABLE emp (
      emp_id NUMBER
, first_name VARCHAR2(128)
, last_name VARCHAR2(128)
) COMPRESS [FOR DIRECT_LOAD OPERATIONS];
```

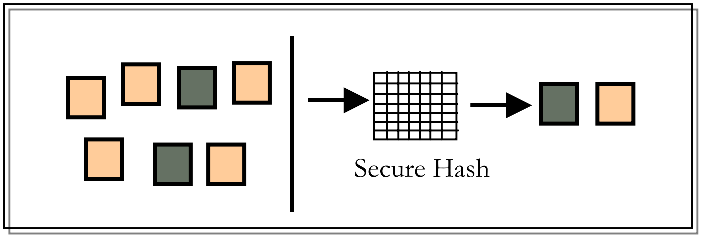
Compression for Unstructured Data

SecureFiles, a new feature in Oracle Database 11g, offers a 'best-of-both-worlds' architecture for storing unstructured content, such as documents, spreadsheets and XML files. SecureFiles is specifically engineered to deliver high performance for file data comparable to that of traditional file systems while retaining the advantages of the Oracle database. SecureFiles is designed as a superset of the ANSI standard LOB and offers easy migration from old LOBs or BasicFiles. With SecureFiles, organizations can now manage all relational data and associated file data in Oracle using a single security/audit model, a unified backup & recovery process, and perform seamless search across all information. The Advanced Compression Option of Oracle Database 11g includes technologies that drastically reduce the storage footprint of SecureFiles data.

SecureFiles Deduplication

It is extremely common for applications to store exact replicas of files. A common example is an email application where multiple users may receive the same attachment. SecureFiles Deduplication is an intelligent technology that eliminates duplicate copies of SecureFiles data. Oracle 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 SecureFiles 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 had used SecureFiles with Deduplication, it would have stored the 1MB attachment just once. That's a 90% savings in storage requirements. In addition to the storage savings, SecureFiles Deduplication also increases application performance. Specifically, write and copy operations are much more efficient since only references to the SecureFiles image are written. Further, read operations may improve if duplicate SecureFiles data already exists in the buffer cache.

Figure 4. SecureFiles Deduplication



How to Enable SecureFiles Deduplication

Deduplication can be enabled for SecureFiles as below:

```
CREATE TABLE images (
image_id NUMBER,
image BLOB)
      LOB(image) STORE AS SECUREFILE
(TABLESPACE lob_tbs DEDUPLICATE);
```

SecureFiles Compression

The Advanced Compression Option of Oracle Database 11g provides yet another mechanism to control the size of your SecureFiles data. In addition to Deduplication discussed earlier, SecureFiles Compression utilizes industry standard compression algorithms to further minimize the storage requirements of SecureFiles data. Compression of typical files, such as documents or XML files, experience a reduction of 2 to 3 times in size. Using built-in intelligence, SecureFiles Compression automatically avoids compressing data that would not benefit from compression – for instance a document that was compressed via a 3rd party tool before being inserted into the database as a SecureFile.

Currently, two levels of compression are supported – with the higher level achieving greater compression but requiring more CPU utilization. The typical CPU overhead of SecureFiles Compression is between 3% and 5%. Applications are still able to perform random reads and writes on compressed SecureFiles data since the compressed data is broken down into small chunks of data. This can vastly improve performance when compared with compressing entire files before inserting them into the database.

How to Enable SecureFiles Compression

The example below shows how compression can be enabled for SecureFiles:

```
CREATE TABLE images (
image_id NUMBER,
image BLOB)
      LOB(image)STORE AS SECUREFILE
(TABLESPACE lob_tbs COMPRESS);
```

Compression for Backup Data

In addition to compressing data stored inside the database, Oracle 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 on the other hand is used to perform a "logical" backup by offloading data from one or more tables into a flat file. Oracle Advanced Compression includes the capability to compress the backup data generated by either of these tools.

Data Pump Compression

The ability to compress the metadata associated with a Data Pump job is provided in Oracle Database 10g Release 2. In Oracle Database 11g, this compression capability has been extended so that table data can be compressed on export. Data Pump compression is an inline operation, so the reduced dump file size 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 uncompress a dump file before importing it. The compressed dump file sets are automatically decompressed during an import operation without any additional steps by the Database Administrator.

In the following compression example from the Oracle sample database, the OE and SH schemas were exported while simultaneously compressing all data and metadata. The dump file size was reduced by 74.67%.

Three versions of the gzip (GNU zip) utility and one UNIX compress utility were used to compress the 6.0 MB dump file set. The reduction in dump file size was comparable to Data Pump compression. Note that reduction in dump file size will vary based on data types and other factors.

Full Data Pump functionality is available using a compressed file. Any command that is used on a regular file will also work on a compressed file. Users have the following options 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.

For more information about Oracle Data Pump, please visit
http://www.oracle.com/technology/products/database/utilities/index.html

How to Enable Data Pump Compression

The example below shows how compression can be enabled for Data Pump:

```
expdp hr FULL=y DUMPFILE=dpump_dir:full.dmp COMPRESS;
```

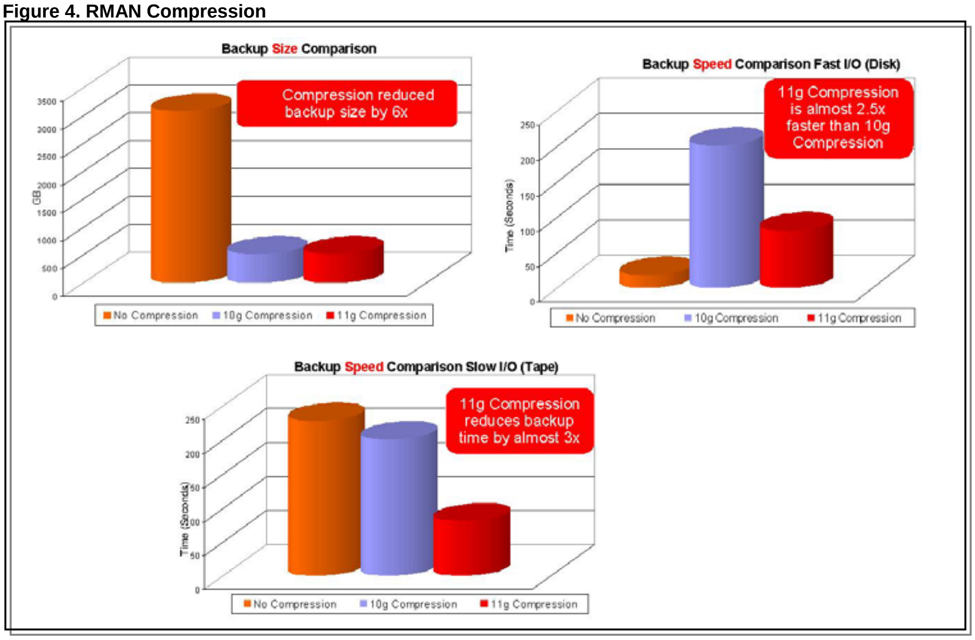
Recovery Manager Compression

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. Recovery Manager (RMAN), Oracle's backup and recovery utility, introduced compression capabilities in Oracle Database 10g. RMAN Compression provides a dramatic reduction in the storage required for backups. Due to RMAN's tight integration with Oracle Database, backup data is compressed before it is written to disk or tape and doesn't need to be uncompressed before recovery – providing an enormous reduction in storage costs. However, backup performance is impacted due to the extensive compression ratio, resulting in longer backup windows.

Oracle Advanced Compression introduces new RMAN Compression capabilities that improve RMAN performance while still drastically reducing the storage requirements for backups. Based on the industry standard ZLIB compression algorithm, RMAN compressed backups are up to 40% faster than compressed backups in Oracle Database 10g. Oracle achieves this drastic performance improvement while marginally reducing the compression ratio by less than 20%. Fast RMAN Compression is a perfect solution for incremental backups taken during normal business hours.

The charts below depict impact of compression on RMAN backup size and performance. These tests were conducted on a real world ERP database.

Figure 4. RMAN Compression



Backup Size Comparison

Compression reduced backup size by 6x

Backup Speed Comparison Fast I/O (Disk)

11g Compression is almost 2.5x faster than 10g Compression

Backup Speed Comparison Slow I/O (Tape)

11g Compression reduces backup time by almost 3x

These charts demonstrate that Advanced Compression option of Oracle Database 11g provides much faster compressed RMAN backups when compared with Oracle Database 10g. This helps in keeping the backup window small while reducing the size of backups.

How to Enable RMAN Compression

Syntax for Fast RMAN compression is as below:

```
RMAN> CONFIGURE COMPRESSION ALGORITHM 'zlib';
```

RMAN compression can be done as shown below:

```
RMAN> backup as COMPRESSED BACKUPSET database
archivelog all;
```

Compression for Network Traffic

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 are used to transfer this redo data to the standby site(s).

Network or standby server outages can prevent redo data from being transported to the standby server. When the outage is resolved, Oracle automatically performs redo gap resolution by transporting all redo data needed to synchronize the standby database. Oracle Advanced Compression introduces the capability to compress redo data as it is sent over the network during redo gap resolution. Through this compression network bandwidth is maximized to increase the gap resolution throughput. Gap resolution can be up to two times faster with compression – ensuring that the standby database is quickly synchronized and High Availability is achieved.

For more information about Oracle Data Guard, please visit
http://www.oracle.com/technology/deploy/availability/index.html

CONCLUSION

The explosion in data volume being experienced by enterprises introduces significant challenges. Companies must quickly adapt to the changing business landscape without impacting the bottom line. IT managers need to efficiently manage their existing infrastructure to control costs, yet continue to deliver extraordinary application performance.

The Advanced Compression Option of Oracle Database 11g provides a robust set of compression capabilities that enable IT managers to succeed in this complex environment. Leveraging the Advanced Compression Option, enterprises can efficiently manage their increasing data requirements throughout all components of their data center – minimizing costs while continuing to achieve the highest levels of application performance.

Oracle Database 11g
Oracle Advanced Compression
April 2008
Author: William Hodak
Contributing Authors: Carol Palmer, Timothy Chien, Sushil Kumar, Ajeet Singh

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com