



An Oracle White Paper
May 2011

Optimizing Storage in SAP with Oracle Database 11g Release 2

Introduction.....	1
Optimizing Storage in SAP with Oracle Database 11g Release 2	2
Storage Efficiency	2
Index Compression and Bitmap Indexes.....	3
Oracle Advanced Compression	3
OLTP Table Compression.....	4
SecureFiles Compression and Deduplication.....	4
Backup Compression.....	5
Network Compression.....	6
SAP Benefits with Advanced Compression	6
Implementation Overview for Compression in SAP Environments .	6
SAP R/3 and BW Tables -- Compression Results Examples	7
Additional SAP Compression/Storage Capabilities	8
Deferred Segment Creation	8
Dictionary Only Add Column	8
Unicode Charactersets	9
Additional SAP Compression/Storage Benefits	9
Conclusion.....	10

Introduction

Using actual customer data, Oracle Database 11g Release 2 compression and space optimizations reduced the overall size of SAP systems from 4.7 TB to 1.9 TB for R/3 (ECC6.0), from 1.4 TB to 0.5 TB for BW (7.0) and from 0.9 TB to 0.3 TB for CRM (7.0). As this customer data shows, the benefits from compression can be significant, yet some organizations still have uncertainty regarding the performance impact that compression may have on their systems. Oracle testing, using compression on a SAP R/3 (ECC 6.0) system under production load, has shown that Oracle's revolutionary compression technology provided storage savings with no significant overhead in CPU consumption, required 60% less physical reads and produced a 10% better database cache rate.

Exponential increases in data volumes in recent years have put enterprise IT infrastructures under severe pressure – from a cost, performance, scalability and manageability perspective. It has become imperative to employ more efficient ways of storing and managing data to meet the growing demands being placed on IT systems. Drastic increases in storage volumes are evident in all types of applications, however, enterprise applications have proven to be among the fastest growing applications.

Businesses also have growing requirements to run applications and analytics faster and faster in spite of the growth in storage volumes. The capabilities provided by Oracle's Advanced Compression Option (ACO) enables many customers to run the database in an in-memory mode by being able to fit the entire working set of the application in memory after 2x to 3x compression – giving applications a significant leap in performance. And given the growth experienced by today's databases, operations like backups are increasingly seen as taking longer to perform – Oracle's Advanced Compression Option can help make those operational tasks faster, thereby providing both storage efficiency + performance as benefits.

Even prior to Oracle Database 11g Oracle had already been a world leader in storage efficiency for customer data – Oracle's database size is generally significantly smaller than other databases because of the key attributes of Oracle's data representation on disk (see *Storage Efficiency* section below). This has helped Oracle databases, especially under SAP, to be more efficient in terms of data storage compared to other vendors' databases and now with Oracle Database 11g Release 2 Oracle has again set the industry standard by becoming a leader in compressed technology.

This paper discusses the Oracle Database 11g Release 2 capabilities that enable database administrators to control the ever-expanding storage volumes of their SAP databases and quantifies how those features will benefit the underlying database in a SAP environment.

Optimizing Storage in SAP with Oracle Database 11g Release 2

Storage Efficiency

Even before applying compression, there are significant advantages in the way data is internally stored in an Oracle database. Oracle's storage structures focus on efficiency and utilize only the minimum required space for each data type. Numeric data is stored with variable length in Oracle - NUMBER data type - which means that Oracle uses only the minimum required space to store the NUMBER data type. Compare this to other DBMSes, which often store numeric data using a fixed length format - DECIMAL data type. A DECIMAL(16,2) data type, for example, will require 16 digits in other DBMSes -- no matter the actual number of digits.

Similarly, when storing variable length character data using VARCHAR, there is often more overhead with other DBMSes than with Oracle Database. Some DBMSes uses four additional bytes to store information about VARCHAR data, irrespective of the size of actual data. For example, a VARCHAR(10) field containing the string 'DB' could require a total of 2+4=6 bytes of storage in other DBMSes. Oracle Database's overhead for storing such data can be as small as one byte, depending on the actual data size. In this example, Oracle Database will use only one additional byte to store information and will require 2+1=3 bytes of storage.

To understand the impact of Oracle's efficient internal storage structures on real world data, it is important to remember that even without compression there are significant differences in the way data is internally stored in Oracle compared to other DBMSes. Because of this greater storage efficiency, Oracle requires less storage space, out-of-the-box, for the same data compared to other DBMSes -- even without (explicit) compression. Depending on the table structure, this difference can easily get extended to a point where other DBMSs use 40-50% more space than Oracle Database to store an equivalent dataset in a real world setting.

Index Compression and Bitmap Indexes

Index compression is part of the Enterprise Edition so no additional options or licenses are required. Index compression allows compression of key column values in an index or index-organized table, thereby reducing the storage for repeated values. Starting with Oracle Database 10.2.0.2, SAP certified the use of index compression to save disk space for indexes and reduce total database size on disk. Customer experiences show that up to 75% less disk space (compared to using non-compressed indexes) is needed for key compressed indexes. Real world example: The size of the index 'GLPCA~1' was reduced from 18GB to 4.5GB. Even after full database reorganization has taken place, an additional 20% of total disk space reduction for the whole database can be achieved using index compression.

Bitmap indexes are used in SAP BW to model star and snow flake schemas in the database. Through the use of bitmap indexes Oracle is able to implement highly efficient star transformation techniques for SAP BW reports. Bitmap indexes are compressed by definition and do not require compression to be turned on separately. A bitmap index uses a bitmap (or bit vector) for each key value instead of a list of row identifiers. As the bitmap representation is very efficient from a storage perspective, bitmap indexes can save significant space over other index types, especially for low and medium cardinality data, and also provide significant performance improvements. For columns where each value is repeated hundreds or thousands of times, a bitmap index typically could be less than 25% of the size of a regular B-tree index.

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 storage costs. It allows IT administrators to significantly reduce their overall database storage footprint by enabling compression for all types of data – be it relational (OLTP table compression), unstructured (SecureFiles Compression and Deduplication), network (Data Guard Redo Transport) or backup (RMAN and Data Pump) data.

Although storage cost savings are 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 a number of components of your IT infrastructure, including memory and network bandwidth. A brief description of each Advanced Compression capability is outlined below, followed by an example benefit of implementing that feature in an SAP environment.

OLTP Table Compression

Oracle's OLTP Table Compression uses a unique compression algorithm specifically designed to work with OLTP applications. The algorithm works by eliminating duplicate values within a database block, even across multiple columns. The compression ratio achieved in a given environment depends on the nature of the data being compressed, specifically the cardinality of the data. In general, SAP customers can expect to reduce their storage space consumption by a factor of 2x to 4x by using the OLTP Table Compression feature. That is, the amount of space consumed by compressed data will be two to four times smaller than that of the uncompressed data.

The benefits of OLTP Table Compression go beyond just on-disk storage savings. One significant advantage is Oracle's ability to read data in 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 will improve due to the reduction in I/O since Oracle will have to access fewer blocks. Furthermore, the buffer cache will become more efficient by storing more data without having to add memory.

SecureFiles Compression and Deduplication

SecureFiles, a feature introduced in Oracle Database 11g Release 1, offers a 'best-of-both-worlds' architecture for storing semi-structured and 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 data type and offers easy migration from existing BasicFile LOBs, the precursor to SecureFiles.

Securefiles are important for the following reasons:

- Very important for SAP Applications like CRM, Portal and PI as they use the LOB Datatype a lot for storing documents, contracts, workflow information and etc...
- Significantly faster access times compared to LOBs in SAP environments -- improved performance over Basicfiles and LONGs for larger LOBs
- Improved Oracle RAC scalability over Basicfiles -- increased transaction throughput on SAP cluster tables especially with Oracle RAC
- Prerequisite for compression and encryption
- Used by all SAP products in the following way:
Replacing LONG and LONG RAWs used in many SAP tables such as SAP cluster tables in SAP ERP

Advanced Compression has two storage optimization features that can be leveraged with SecureFiles. The first feature, SecureFiles Deduplication, is an intelligent technology that eliminates duplicate copies of SecureFiles data. Oracle stores one copy of the SecureFiles LOB and replaces the duplicate copies with references to the first copy. Consider a content management application where 10 users store an identical 1MB document in the system. Without SecureFiles Deduplication, the system would store one copy of the file for each of the 10 users – requiring 10MB of storage. If the application uses SecureFiles with Deduplication, it will store the 1MB attachment just once. That’s a 90% savings in storage requirements.

The second feature, SecureFiles Compression, utilizes industry standard compression algorithms to further minimize the storage requirements of SecureFiles data. With SecureFiles compression typical files, such as documents or XML files, experience a reduction of 2x to 3x 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 SecureFiles file. With three levels of compression available (LOW, MEDIUM and HIGH), users can determine the optimal storage savings and compression CPU overhead for their environment. SecureFiles compression can result in huge storage savings for applications storing unstructured data inside the database.

Backup Compression

In addition to compressing data stored inside the database, Oracle Advanced Compression also includes the capability to compress backup 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. Oracle Advanced Compression includes the capability to compress the backup data generated by both of these tools.

Oracle 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 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. Similarly, the data and metadata generated by Data Pump exports can be compressed with Advanced Compression. Data Pump compression is an inline operation, so the reduced dump file size means a significant savings in disk space. But unlike operating system or file system compression utilities, there is no need to uncompress a Data Pump dump file before importing it.

Network Compression

Oracle Data Guard provides the management, monitoring, and automation software infrastructure needed to create, maintain, and monitor one or more standby databases to protect enterprise data from failures, disasters, errors and data corruption. 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). 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. As of Oracle Database 11g Release 2, 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).

SAP Benefits with Advanced Compression

Using various customers' SAP data, Oracle conducted tests using Advanced Compression features to estimate the total storage savings in a SAP environment. Features such as OLTP table compression, index prefix compression, RMAN compression and Data Pump compression yielded significant savings. The table below outlines the extraordinary benefits of Advanced Compression for SAP.

Object Type	Compression Feature	Original Size	Compressed Size	Compression Factor
Tables/Indexes	OLTP Table Compression/Index Prefix Compression	950 GB	334 GB	2.8x
Data Pump Exports	Data Pump Compression	973 GB	156 GB	6.2x
RMAN Compression	RMAN Compression, LEVEL=HIGH	2.8 TB	667 GB	4.2x

Source: Oracle Tests on Customer Databases

Implementation Overview for Compression in SAP Environments

Below are some basic implementation guidelines regarding how to approach an Oracle Database 11g Release 2 upgrade in terms of implementing compression technologies in SAP environments

for the purpose of storage optimization:

1. Start with Index Compression

- a) Improves R/3 performance the most as all database access in R/3 environments is index based, especially index range scans
- b) Up to 20% disk space reduction for complete database

2. Implement OLTP Table Compression

- a) Reporting queries and OLTP run faster by being able to hold more data in the buffer cache and reducing disk I/O
- b) Provides very good disk space savings

3. Implement SecureFiles Compression

- a) Very important for SAP Applications like CRM or XI who store large (uncompressed) amount of data in LOBs

4. RMAN Backup Compression

- a) Additional disk space savings (even if database tables and indexes are already compressed)

SAP R/3 and BW Tables -- Compression Results Examples

R/3 Tables	Uncompressed (size in GB)	OLTP Compressed (size in GB)
MSEG	80	19 (4.2x compression)
BSIS	78	17 (4.6x compression)
COEP	74	21 (3.5x compression)

Source: Oracle Tests on Customer Databases

BW Tables	Uncompressed (size in GB)	OLTP Compressed (size in GB)
/BIC/B0000689000	37	5 (7.4x compression)
/BIC/B0000564000	34	8 (4.3x compression)
/BIC/B0000066000	23	5 (4.6x compression)

Source: Oracle Tests on Customer Databases

Additional SAP Compression/Storage Capabilities

Deferred Segment Creation

In Oracle Database 11g Release 2, non-partitioned heap-organized tables can take advantage of a new feature called Deferred Segment Creation (with 11.2.0.2 this capability also works for partitioned objects (tables, indexes)). As the name implies, the segment creation for these tables is deferred until the first row is added to the table. In addition, LOB segments, and any associated indexes for that table, are deferred until the first row is inserted.

The main advantage of deferred segment creation is storage savings for application tables that are automatically created but never populated with data. In addition to the space savings, deferred segment creation increases the speed of SAP system installation since it eliminates the space allocation operations for empty objects during an SAP installation. This type of behavior is extremely common in SAP environments, where typical installations create thousands of tables -- many of which are never populated with data.

Queries on data dictionary tables are also faster. SAP heavily queries Oracle's data dictionary tables as part of its built-in space management processes. In many SAP implementations, up to 70% of tables are created but never populated. Therefore, deferred segment creation can significantly benefit SAP customers through storage savings, reduced installation time and performance improvements for querying data dictionary tables.

Dictionary Only Add Column

Prior to Oracle Database 11g, adding new columns with DEFAULT values and a NOT NULL constraint to an existing table required both an exclusive lock on the table and the default value to be stored in all existing records. In Oracle Database 11g, the database optimizes the resource usage and storage requirements for this same operation.

Specifically, default values of columns are maintained in the data dictionary for columns specified as NOT NULL. With this new feature, adding new columns with DEFAULT values and NOT NULL constraints only requires the necessary metadata to be stored in the data dictionary and no longer requires the default value to be stored physically in all existing records.

Adding columns in this manner is very common for both the SAP BW applications, and SAP upgrades. In general the benefits of dictionary only add column include:

- Sub-second schema modifications and no space usage - regardless of the size of the table or added columns
- While tables of all sizes benefit from this feature, adding columns to large tables in this fashion is dramatically faster
- Significantly reduces disk space usage
- The process of adding columns in SAP BW triggered through cube changes is now more than 20x faster. Cube changes are very frequently triggered by BW users when additional key figures are needed in SAP BW reports.
- Similar performance improvement can be seen for SAP upgrades as additional columns are added for tables.

Unicode Charactersets

Oracle uses a variable length representation on disk for multi-byte character sets; SAP uses the UTF-8 character set within the Oracle Database for SAP Unicode systems. The variable length representation is a significant space advantage, compared to other database vendors, which reportedly have to use a fixed 2 Byte character set for SAP Unicode systems. About 40% of disk space is saved by using the variable length UTF-8 character set for SAP enabled Unicode systems compared to the fixed multi-byte representation of other databases.

Unicode is becoming very important for the following reasons:

- Upcoming new SAP versions are Unicode only
- All Java based SAP products require Unicode

Additional SAP Compression/Storage Benefits

Using various customers' SAP data, Oracle conducted tests using the additional Oracle capabilities to estimate the storage savings in a SAP environment. The table below outlines the additional estimated storage benefits provided by these capabilities:

Oracle Capability	Benefits	Comments
Deferred Segment Creation	Approximately 1% of total database disk space is saved on R/3 systems and 5-7% for BW systems	While disk savings is a benefit, the main advantages of deferred segments is the performance increase for queries against the data

		dictionary and the reduced installation times of SAP system as no space allocations are required for empty objects
Bitmap Indexes	Through the use of disk-based bitmap indexes 25% of total database disk space is saved	With bitmap indexes Oracle is able to implement highly efficient star transformation techniques for SAP BW reports
Dictionary Only Add Column	Approximately 2% of total database disk space is saved	Adding new columns with DEFAULT values and NOT NULL constraints only requires the necessary metadata to be stored in the data dictionary and no longer requires the default value to be stored physically in all existing records
Unicode Charactersets	About 40% of disk space is saved by using the variable length UTF-8 character set for SAP enabled Unicode systems	Upcoming new SAP versions are Unicode only

Conclusion

To quantify the results in an SAP environment, customer data was used to compare the storage consumption before and after implementing the individual features. Using this actual customer data, Oracle 11g Release 2 compression and space optimizations reduced the overall size of their SAP systems from 4.7 TB to 1.9 TB for R/3 (ECC6.0), from 1.4 TB to 0.5 TB for a BW (7.0) and from 0.9 TB to 0.3 TB for CRM (7.0).

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.

Built-in Oracle database capabilities including: Index Compression, Deferred Segment Creation,

Dictionary Only Add Column, Storage Efficiency, Bitmap Index's and Unicode Charactersets provide SAP users with significant advantages in storage savings, improved data dictionary query performance and faster SAP installation times.

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. Using 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.

Additional Information

- **SAP AMI Performance White Paper** (includes use of Advanced Compression Option) ¹ (see [here](#))
- **Customer Use Examples:**
 - Oracle Advanced Compression at SHARP Japan** (page 50) ² (see [here](#))
 - Oracle Advanced Compression at TVS Motor Company** (page 54) ³ (see [here](#))

¹ <http://www.oracle.com/technetwork/database/options/compression/ami-performance-359599.pdf>

² <http://www.oracle.com/us/solutions/volume20-sapnewsletter-eng-360057.pdf>

³ <http://www.oracle.com/us/solutions/volume20-sapnewsletter-eng-360057.pdf>



Optimizing Storage in SAP with Oracle
Database 11g Release 2
May 2011
Author: Gregg Christman
Contributing Authors: Jan Klokkers, Amit
Ganesh and Kevin Jernigan

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



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2010, 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.

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

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. 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. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0410

SOFTWARE. HARDWARE. COMPLETE.