ORACLE
E-BUSINESS SUITE

An Oracle White Paper
May 2010

Oracle E-Business Suite Release 12.1 with Oracle Database 11g Advanced Compression

ORACLE

Executive Overview

The Oracle Database 11g Advanced Compression Option (ACO) introduces a comprehensive set of compression capabilities designed to help customers maximize resource utilization and reduce costs thereby reducing the total cost of ownership (TCO). It allows IT administrators to significantly reduce their overall database storage footprint by enabling compression for all types of data including relational (tables), unstructured (files), and backup 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 all components of your IT infrastructure including memory and network bandwidth.

The volume of data being retained on-line increases the need for Life Cycle Management. Typically historical data is partitioned and moved to low-cost storage. Advanced Compression reduces the storage costs further ensuring that application performance does not degrade unacceptably. Oracle has saved more than 68 TB across its Oracle E-Business Suite implementation.

Introduction

This white paper details tests performed with Oracle E-Business Suite Release 12.1 running on Oracle Database 11g Release 1 with Advanced Compression using a mix of online and batch workloads to validate and quantify the benefits of Advanced Compression. The tests described in this paper focus on the OLTP Table Compression feature of Advanced Compression. Representative batch and online workloads were selected, such as the Order-To-Cash and Payroll Batch flows in addition to various Self-Service and Forms based online processes. The benchmark information includes details of the workloads, the tables chosen for compression, and test descriptions. The test results are analyzed and evaluated and used to provide a set of recommendations.

Advanced Compression Overview

Oracle introduced basic table compression in Oracle 9i, which compresses data loaded via bulk operations using direct-path insert. OLTP Table Compression in Oracle Database 11g uses a unique compression algorithm specifically designed to work with OLTP applications. The algorithm works by eliminating duplicate values within a database block and even across multiple columns. Compressed blocks contain a structure called a symbol table that maintains compression metadata as shown diagrammatically in Figure 1.

In addition to the benefit of saving storage with OLTP Table Compression, Oracle Database can read compressed blocks directly without decompression either on disk or in memory. 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 as fewer blocks are accessed for a given set of rows. Figure 1 shows an example of database block compression, which can be enabled at table or partition level.

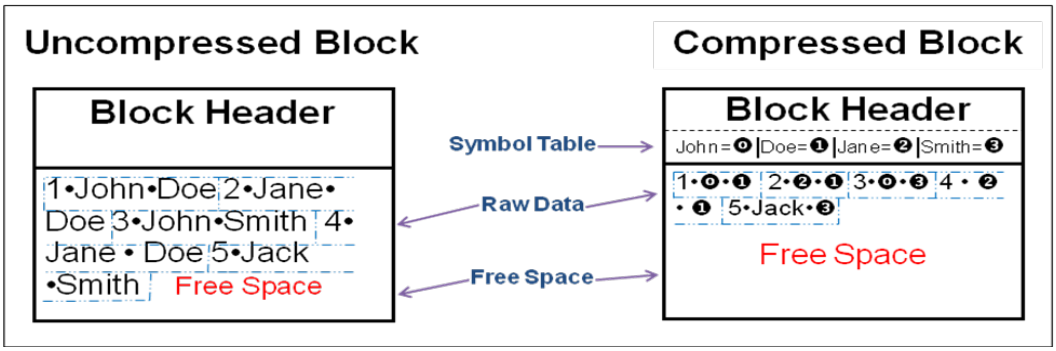

Figure 1. Block Compression

For DML operations Oracle Database compresses blocks in batch mode rather than compressing data each 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, the entire block contents are compressed, as shown in Figure 2.

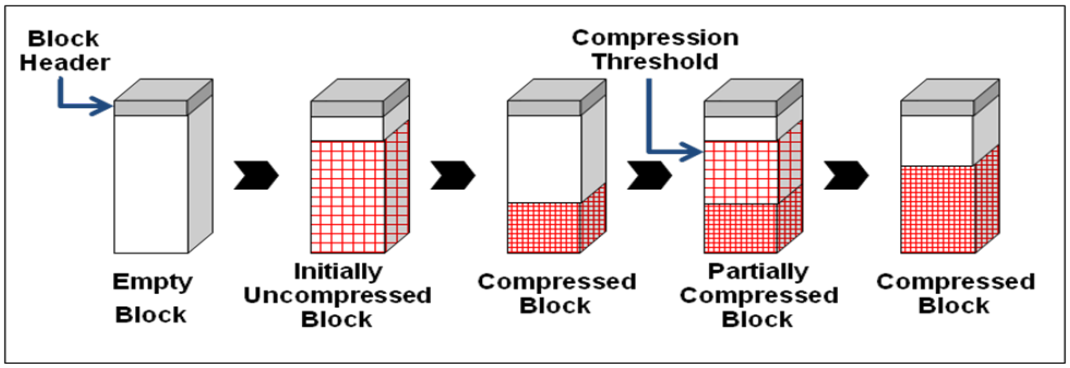

Figure 2. Block Compression

As of 11gR1, update operations incur some overhead as updated rows have to be decompressed before writes and therefore the update frequency is one factor to consider when choosing tables or partitions to be compressed. Read-only operations should benefit from a performance improvement while update-intensive operations will incur some overhead. However, this is expected to balance out given the common read-intensive I/O patterns seen in most OLTP applications.

Testing

This section details the test environment and the application flows used in each scenario. It then describes the testing methodology and discusses the results.

Test Environment

The hardware and software details for the test environment are as follows:

- Oracle Database 11g Release 1 (11.1.0.7) with recommended Advanced Compression and DML/Space Layer optimization patches, which are listed in Appendix A
- Oracle E-Business Suite 12.1.1 VISION Environment with data expanded for the benchmark
  - Database Initialization Parameter Settings for Oracle Applications Release 12 from My Oracle Support Knowledge Document 396009.1. No parameters were changed for this benchmark.
- Database Server Hardware:
  - Linux 2.6
  - 16 GB RAM
  - 2 Dual Core XEON CPU's @ 2.3 GHz
  - Database Storage: NetApp Filers

The VISION Database used for the tests was approximately 340 GB

- 180 GB (uncompressed) of allocated space for following objects
  - 25 GB for System/Undo/Temp/SysAux
  - 50 GB for Application Indexes
  - 82 GB for Application Tables
  - 23 GB for other object types

Test Scenarios

The following Oracle E-Business Suite flows were tested:

- **Batch Flows**
  The following batch jobs are CPU and I/O intensive:
  - **Payroll Process** – Consists of 7 programs processing 10,000 employees, using 4 workers. This process contains significant row-by-row processing.
  - **Order-To- Cash** – Consists of 12 programs from the Supply Chain and the Financials product families, processing, invoicing, accounting and posting data for 5,000 order headers and 50,000 order lines, using 4 workers. This process is a mix of row-by-row and bulk processing.
- **Online Flows**
  The following read-intensive online scenarios assess the impact to end-user response times:
  - **Self Service** – 100-user test of 7 Self Service actions related to expenses, timecards, and payslips.
  - **Oracle Forms** – 120-user test of 16 Purchasing actions related to creating, approving, and searching purchase orders and invoices.

Testing Methodology

The following briefly summarizes the testing methodology used to compare both the storage savings and performance impact of Advanced Compression with Oracle E-Business Suite.

Step 1: Establish Baseline

All of the test scenarios were executed prior to enabling Advanced Compression in order to establish a performance baseline.

Step 2: Enable OLTP Table Compression

All tables accessed during the baseline tests were identified. Of these, the largest 200 tables were selected (listed in Appendix B) for OLTP Table Compression. Prior to enabling OLTP Table Compression, each table was first reorganized by using the *ALTER … TABLE MOVE* command, and the baseline size of the table was recorded. Then, OLTP Table Compression was enabled on each table using the *ALTER TABLE … MOVE COMPRESS FOR ALL OPERATIONS* command. The delta between the *MOVE* and *COMPRESS* was recorded.

Step 3: Run Test Scenarios with Advanced Compression

All the test scenarios were run with OLTP Table Compression enabled. Storage consumption and performance statistics were collected including Automatic Workload Repository reports, host CPU statistics, batch program runtimes, and end-user response times. The results of each test were validated by comparing the expected and actual results including, for example, the number of orders or employees processed.

Test Results

Tables with more than 255 columns or that include columns with the LONG data type cannot be compressed using OLTP Table Compression as it is not supported. The test environment contained 38 tables totaling 1.4 GB that could not be compressed due to these restrictions.

The focus of this benchmark was OLTP Table Compression. Columns with BasicFiles LOBs were not converted to SecureFiles LOBs and therefore not compressed. Indexes were not compressed. Clearly the total savings would have been greater if these objects had been compressed.

Storage Savings

As discussed, each table was initially reorganized to ensure that the space saving observed from OLTP Table Compression was attributed only to the compression technology. Table 1 shows the storage savings achieved using Advanced Compression.

TABLE 1. STORAGE SAVINGS

DESCRIPTION	STORAGE UTILIZED	STORAGE SAVED	PERCENT SAVINGS
Uncompressed Tables	21.9 GB	N/A	N/A
Uncompressed Tables After Reorganization	21.8 GB	110 MB	0.5
Tables with OLTP Table Compression	6.9 GB	14.9 GB	**68**

Observations

The space savings from reorganizing the tables prior to enabling OLTP Table Compression was insignificant at 110 MB. However, this was a new install and the amount of space saved is likely to be far more substantial on an OLTP system that has been operational for some time.

OLTP Table Compression reduced the size of the tables from 21.8 GB to 6.9 GB as shown in Table 1. This equates to a 68% reduction in storage utilization, which is approximately a 3:1 compression ratio.

The storage savings achieved by implementing OLTP Table Compression can vary across tables depending on the data. Tables that store similar data typically achieve higher compression ratios. The tables in this test environment achieved compression ratios ranging from better than 5:1 to less than 2:1. The following is the distribution of compression ratios in the VISION test environment with additional data for this benchmark:

- 36 tables with a compression ratio > 5:1
- 27 tables with a compression ratio between 4:1 and 5:1
- 90 tables with a compression ratio between 2:1 and 4:1
- 40 tables with a compression ratio < 2:1

Performance Results

This section analyzes each of the individual test scenarios.

Self Service 100 User Test

Transaction performance improved significantly with Advanced Compression in this test scenario and response times improved between 14% and 36% as shown in Table 2.

TABLE 2. SELF SERVICE TESTS RUNTIMES

TRANSACTION	BASELINE (MINS)	ADVANCED COMPRESSION (MINS)	IMPROVEMENT %
Cash Expense Report	0.342	0.273	20
Cash Expense Submit	0.628	0.437	30
Credit Card Expense Report	0.371	0.320	14
Credit Card Expense Submit	0.634	0.453	29
Timecard Submit	0.123	0.093	24
Timecard Total	3.110	2.524	19
View Payslip	2.444	1.562	36

During the initial run of this test scenario with Advanced Compression, one of the execution plans changed causing a performance regression in the View Payslip screen. Typically, this type of problem can be resolved using a range of techniques including, for example, by gathering statistics, using SQL Profiles, or using SQL Plan Baselines. The results show the performance results after the execution plan was corrected, in this case, by using a SQL Profile.

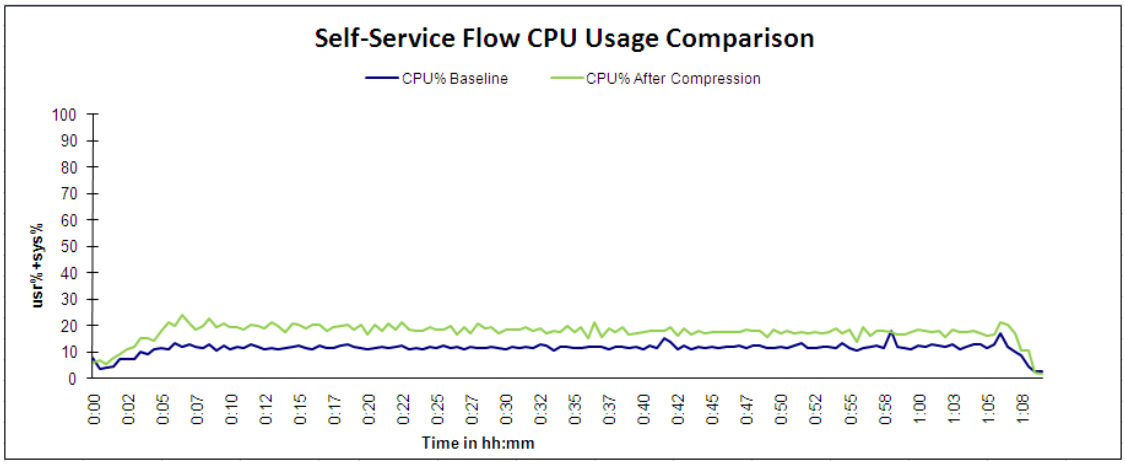

Figure 3 Self-Service Flow CPU Usage Comparison

The graph in Figure 3 shows that average CPU consumption increased by 6.2% with Advanced Compression.

Purchasing - 120 User Test

Advanced Compression improved transaction performance for all but two of the flows in this test scenario and response times varied between -1% and 10% as shown in Table 3.

TABLE 3. PO FLOW TESTS

TRANSACTION	BASELINE (MINS)	ADVANCED COMPRESSION (MINS)	IMPROVEMENT %
PO Approve	0.623	0.594	5
PO Auto-Create	0.223	0.201	10
Create Invoice Total	4.393	4.426	-1
Create Requisition Total	4.928	4.752	4
Find/View Inv Total	2.945	2.901	1
PO Find	0.483	0.459	5
PO Lines	1.366	1.250	8
PO Schedules	0.987	0.942	5
PO Search	0.693	0.703	-1

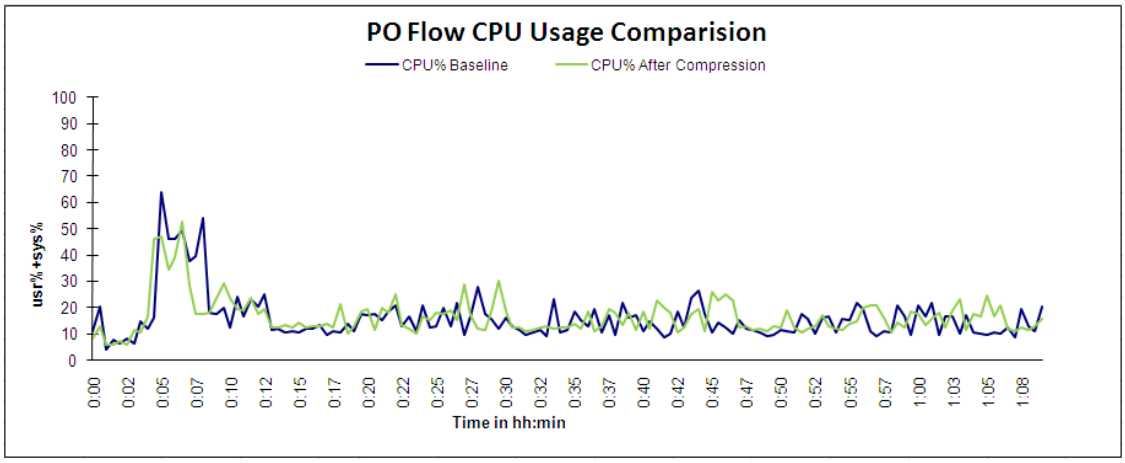

Figure 4 Purchasing Flow CPU Usage Comparison

The graph in Figure 4 shows average CPU consumption increased by 0.7% with Advanced Compression.

Order To Cash Batch Flow

When using Advanced Compression performance improved for all programs except for the Pick Release part of this flow. Elapsed times improved between 2.4% and 33%, but the Pick Release elapsed time increased by 20% due to the high CPU and I/O intensive nature of the program. This is shown in Table 4.

TABLE 4. ORDER TO CASH BATCH FLOW TEST RESULTS

PROGRAM	BASELINE (MINS)	ADVANCED COMPRESSION (MINS)	IMPROVEMENT %
HVOP	6.8	6.78	0.29
Pick Release	21.28	25.58	-20
ITS time	3.87	3.32	14
Inventory Worker	9.08	8.17	10
Auto Invoice	10.43	10.18	2.4
Revenue Recognition	6.02	4.02	33
Accounting	5.52	5.25	5.4
TOTAL	**63.00**	**63.3**	**-0.43**

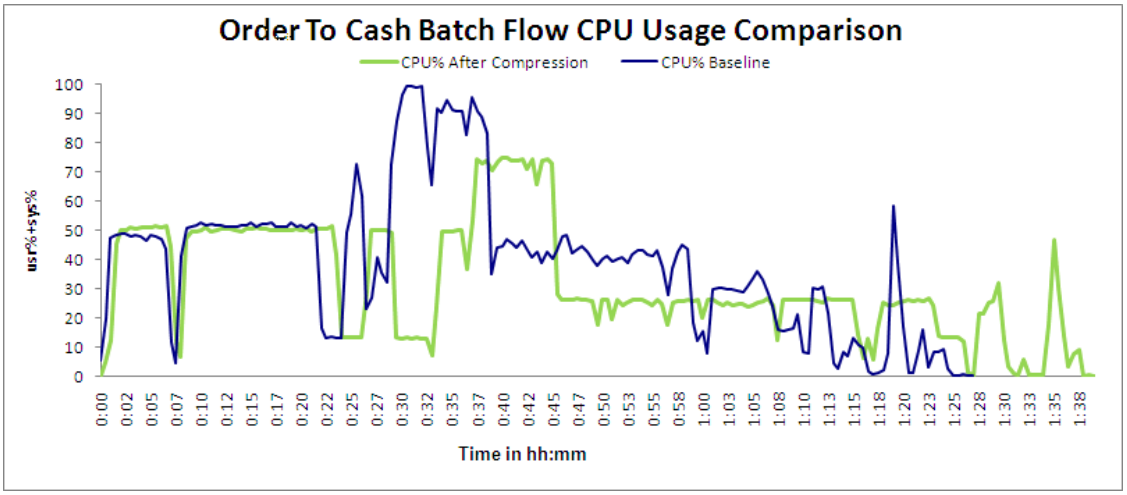

Figure 5 Order To Cash Batch Flow CPU Usage Comparison

The graph in Figure 5 shows that average CPU consumption decreased by 7% with Advanced Compression.

Payroll Batch Flow

Overall performance of the Payroll Batch improved considerably, even though the Payroll Process and Check Writer regressed as shown in Table 5. For the programs where elapsed times improved, the performance increased between 4% and 26% and this compares well with the reduction of between 1% and 9% for the other slower programs. However, note that the actual time difference on the slower programs is only a few seconds and therefore relatively insignificant.

TABLE 5. PAYROLL BATCH FLOW TEST RESULTS

PROGRAM	BASELINE (MINS)	ADVANCED COMPRESSION (MINS)	IMPROVEMENT %
Payroll Process	2.17	2.19	-1
Pre-Payment	0.82	0.79	4
Archiver	7.87	5.79	26
Nacha	0.62	0.59	5
Check Writer	0.43	0.47	-9
Costing	0.69	0.61	12
TOTAL	**12.59**	**10.45**	**17**

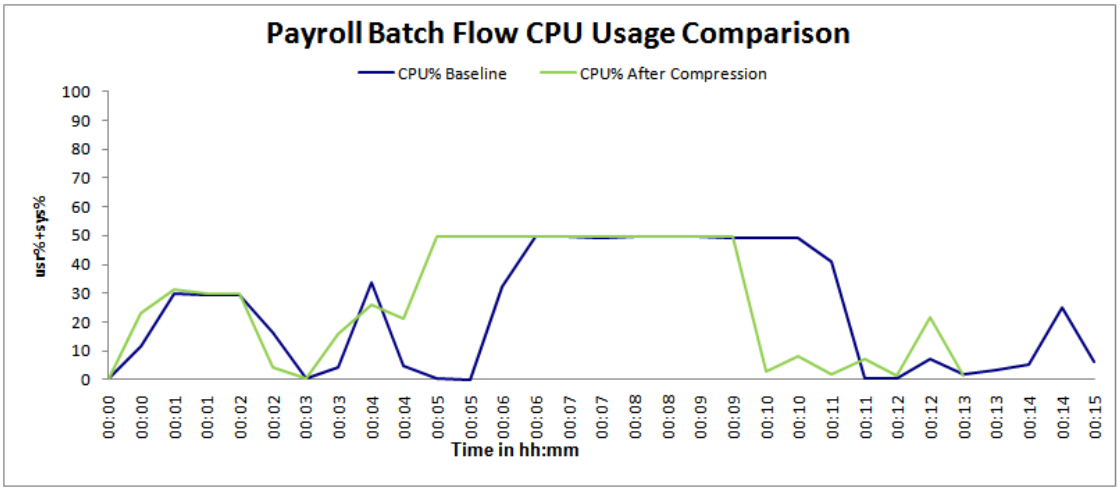

Figure 6 Payroll Batch Flow CPU Usage Comparisons

The graph in Figure 6 shows that average CPU consumption increased by 3.7% with Advanced Compression, however it completed quicker.

Observations

The following observations were made during the tests:

- Some of the SQL execution plans changed reducing performance. These issues could be remedied using Gather Statistics, SQL Profiles, or SQL Plan Baselines. The first approach should always be to Gather Statistics.
- The online workload showed an improvement in response times of up to 30% at the expense of a small increase in CPU of up to 6%. The variance in the results in these tests depends on several factors such as how much DML is performed. For example, the Self-Service flow is more read-intensive so response times improved significantly as expected.
- The Order to Cash Batch scenario is a write-intensive workload and therefore expected to decrease in performance when compared to the other tests. The results show that with Advanced Compression the runtime increased by 0.43% and CPU usage decreased by 7%.
- The Payroll Process Batch workload is both CPU and I/O intensive. With Advanced Compression the runtime reduced by 17% with an additional 3.7% CPU usage.
- There can be increased overhead for Advanced Compression customers running with the initialization parameter *db_block_checking* set to TRUE. Setting this parameter to TRUE causes Oracle to check for corruption on every block change. This block checking process is a function of number of rows per block. With compressed tables, more rows fit into one block and therefore more data will be checked resulting in higher CPU consumption. Database corruption is exceptionally rare and Block Checking is not recommended for Oracle E-Business Suite.

GSI Observations

In 2009 Advanced Compression was implemented with Oracle E-Business Suite Release 12 on the Oracle Global Single Instance (GSI). The database reduced from 17 TB to 13 TB after compressing 110 of the largest tables and indexes, which equates to a total saving of 68 TB across the primary, standby, and associated test systems. This also reduced the storage for development clones and backups. As an aside, Advanced Compression has saved 365 TB on the internal Beehive Email system.

There were two phases of compression in the Oracle production Global Single Instance, which are as follows:

- Phase 1 compressed ~102 of the largest tables (listed in Appendix C), ~700 associated indexes and LOBs on these tables (after migrating to SecureFiles). This freed up 5.6 TB of database storage.
- Phase 2 compressed the next ~160 largest tables (and associated indexes) which freed up a further 620 GB of database storage
- The total GSI production primary space saved is 6.2 TB

Overall, Oracle has saved 68 TB of storage including 2 production databases with corresponding standby, test, and development databases.

Recommendations

The following observations were made during the tests for this whitepaper or during the Advanced Compression with GSI.

- **Index Compression:** As with tables, it is prudent to apply compression to the subset of indexes that provide significant space reduction rather than applying index compression on all indexes. The main reason for this recommended approach is to minimize the execution plan changes. For example, if the size of the index is reduced, the plan may change to perform a full index scan rather than a range scan.
- **Patches:** Review and apply the recommended Advanced Compression Option and DML/Space Layer patches listed in Appendix A. The latter are not directly related to Advanced Compression, but these are highly recommended as they constitute optimizations to the space layer code, which is used by Advanced Compression.
- **SQL Plan Regression**: Expect some SQL execution plan regressions when adopting Advanced Compression. As a first step gather statistics on the objects. If this doesn't resolve the problem then consider using SQL Profiles or SQL Plan Baselines.
- **DML Intensive Operations:** As stated in the overview, Advanced Compression is an ideal choice for saving space with read-intensive operations and can reduce the associated Disk I/O. The performance of DML intensive operations may be reduced, and may incur additional CPU when time is spent compressing data blocks.

- **Row Chaining:** Row chaining is observed on UPDATE intensive tables resulting in a partial loss of compression for the blocks involved in the transaction. In 11gR2, chained rows will be recompressed when the used space threshold is exceeded for the data block.
- **ITL Contention**: Some ITL contention issues resulted in a deadlock on the internal production system. These were caused by a lack of ITL slots when operating under high levels of concurrency. This is not caused by the compression, but is an artifact as inherently storing more rows per block increases the block density. The issue can be resolved by increasing *initrans* on the table, which will increase the ITL slots. Alternatively you could increase *pctfree*, however in this case the space saving will be reduced.
- **High Transaction Tables:** Depending on your application usage, consider excluding high transaction tables such as FND_CONCURRENT_REQUESTS. In exceptionally busy systems, compression is likely to exacerbate the buffer busy waits as there are more rows per block. If you have a high level of concurrent throughput, you will generally find that this is usually among the top segments for buffer busy waits even without compression.
- **SecureFiles LOB Compression:** The LOB data was not compressed for this benchmark. However, LOBs were converted to compressed SecureFiles on the Oracle Global Single Instance with the deduplication feature. Savings totaled 26 TB across the production, standby, and associated test systems.

Planning Your Strategy

When considering compression it is important to review your largest candidate tables. Some of the largest transaction tables in Oracle E-Business Suite have over 255 columns or include Long columns and are therefore not candidates for compression. It goes without saying that it is better to start by compressing your largest tables based on the 80/20 rule, in that 20% of your tables likely consume 80% of the space.

As stated in the overview, update operations will incur some overhead (so performance may be reduced and possibly incur additional CPU) as updated rows have to be decompressed before writes. Therefore this is one of the design choices and requires careful evaluation for tables or partitions subject to frequent updates.

As with tables, index compression can also provide significant storage savings. Oracle Database supports compression of both unique and non-unique indexes. In the case of a non-unique index, all index columns can be stored in a compressed format, whereas in the case of a unique index, at least one index column has to be stored uncompressed.

Compressing large indexes will maximize the storage savings, but compressing indexes on heavily modified tables would negatively impact performance due to contention. So, careful consideration should be given while choosing indexes to compress. Similarly, an optimum number of columns should be chosen. Oracle Database provides an optimum number of columns and corresponding space savings when the index is validated using 'ANALYZE INDEX … VALIDATE STRUCTURE'.

Columns OPT_CMPR_COUNT and OPT_CMPR_PCTSAVE in INDEX_STATS provide the column count and space savings respectively.

Oracle Advanced Compression can significantly reduce the TCO of an Information Lifecycle Management strategy. One very simple approach is to implement time-based partitioning and then move non-current or historical data partitions to low-cost storage. Partitioning is supported for the Oracle E-Business Suite though you may not change an existing integral partitioning scheme. Oracle Advanced Compression can increase the data density either further reducing the need to expand low-cost storage, or it may even substantially defer the infrastructure requirements for secondary storage.

In addition to compressing data stored within the database, Oracle Advanced Compression also includes the capability to compress backed up data. This 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. Oracle Advanced Compression also includes the capability to compress the backup data generated by Data Pump tool.

Appendix A: Recommended Patches

This appendix lists patches recommended for the Oracle Database 11g Advanced Compression Option and optimizations to the DML/Space Layer code, which is used by Advanced Compression.

Oracle Database 11g Advanced Compression Option Patches

TABLE 6. ADVANCED COMPRESSION PATCHES

Patch	Description
8983266	MLR request for 8352309 and 8966797 and 8930565
8930565	Merge request on top of 11.1.0.7.0 for bugs 7281382 7715244 8277580 8287680
8914197	Make 'ALTER SESSION SET DB_BLOCK_CHECKING' a NO-OP
8876094	ORA-00308 in alert logs of our environment
8856478	Perf degradation with SF compression on small LOBS
8834636	Concurrent inserts into compressed table causes high CPU and buffer busy waits
8608377	Alter table move compress with securefile option is slow
8599477	Encountered ORA-04031 with large KDL compression related call memory usage
8503195	Wrong row source time statistics under SQL_TRACE=TRUE
8409203	MLR of 8249087 and 7281382 on 11.1.0.7
8364676	Row cache lock contention when compressing table using ALTER TABLE
8287680	OLTP table compression INSERT performance
8999228	Avoid XDF Utility to change storage definition of compressed index
9011088	Avoid data corruption during insert on a compressed table having > 255 columns

Space Layer Code Patches

TABLE 7. SPACE LAYER CODE PATCHES

Patch	Description
8834636	Concurrent Inserts into Compressed Table Causes High CPU and Buffer Busy Waits
8980613	Significant Buffer Busy Waits - Multiple Sessions Inserting in Compressed Table
9166322	Same Blk Returned Performance Issue (High Logical IOS and Block Changes)
9275072	Buffer Busy Waits Inserting Into Tables
9341448	Buffer Busy Waits and Latch: Cache Buffers Waits When Inserting

Appendix B: Tables Compressed For This Benchmark

Table 8 lists the tables compressed for this benchmark.

TABLE 8. LIST OF COMPRESSED TABLES

AK_ATTRIBUTES	HZ_CUST_ACCOUNTS	PO_HEADERS_ARCHIVE_ALL
AK_ATTRIBUTES_TL	HZ_CUST_ACCT_SITES_ALL	PO_LINES_ALL
AK_REGION_ITEMS	HZ_CUST_SITE_USES_ALL	PO_LINES_ARCHIVE_ALL
AK_REGION_ITEMS_TL	HZ_HIERARCHY_NODES	PO_LINES_INTERFACE
AK_WEB_USER_SEC_ATTR_VALUES	HZ_LOCATIONS	PO_LINE_LOCATIONS_ALL
AP_CARDS_ALL	HZ_ORGANIZATION_PROFILES	PO_LINE_LOCATIONS_ARCHIVE_ALL
AP_CHECKS_ALL	HZ_ORG_CONTACTS	PO_RELEASES_ALL
AP_CREDIT_CARD_TRXNS_ALL	HZ_PARTIES	PO_RELEASES_ARCHIVE_ALL
AP_DBI_LOG	HZ_PARTY_SITES	PO_REQUISITION_HEADERS_ALL
AP_EXPENSE_REPORT_HEADERS_ALL	HZ_PARTY_SITE_USES	PO_REQUISITION_LINES_ALL
AP_EXPENSE_REPORT_LINES_ALL	HZ_RELATIONSHIPS	PO_REQ_DISTRIBUTIONS_ALL
AP_EXP_REPORT_DISTS_ALL	IBY_CREDITCARD	RA_BATCHES_ALL
AP_INVOICES_ALL	IBY_FNDCPT_TX_EXTENSIONS	RA_CUSTOMER_TRX_ALL
AP_INVOICE_DISTRIBUTIONS_ALL	ICX_CAT_ATTRIBUTE_VALUES_TLP	RA_CUSTOMER_TRX_LINES_ALL
AP_INVOICE_LINES_ALL	ICX_CUSTOM_MENU_ENTRIES	RA_CUST_TRX_LINE_GL_DIST_ALL
AP_INVOICE_PAYMENTS_ALL	ICX_SESSIONS	RA_CUST_TRX_LINE_SALESREPS_ALL
AP_PAYMENT_SCHEDULES_ALL	ICX_SESSION_ATTRIBUTES	RA_INTERFACE_SALESCREDITS_ALL
AR_CASH_RECEIPTS_ALL	ICX_TRANSACTIONS	RCV_HEADERS_INTERFACE
AR_CASH_RECEIPT_HISTORY_ALL	JDR_ATTRIBUTES	RCV_SHIPMENT_HEADERS
AR_DISTRIBUTIONS_ALL	JDR_COMPONENTS	RCV_SHIPMENT_LINES
AR_PAYMENT_SCHEDULES_ALL	JDR_PATHS	RCV_TRANSACTIONS
AR_RECEIVABLE_APPLICATIONS_ALL	JTF_RS_RESOURCE_EXTNS	WF_ACTIVITIES
BOM_CALENDAR_DATES	MRP_ATP_SCHEDULE_TEMP	WF_ACTIVITIES_TL
CST_ITEM_COSTS	MRP_RELIEF_INTERFACE	WF_ACTIVITY_ATTRIBUTES
FF_DATABASE_ITEMS	MRP_SO_LINES_TEMP	WF_ACTIVITY_ATTR_VALUES
FF_FDI_USAGES_F	MTL_DEMAND	WF_ACTIVITY_TRANSITIONS
FF_ROUTE_PARAMETER_VALUES	MTL_ITEM_CATEGORIES	WF_COMMENTS
FF_USER_ENTITIES	MTL_MATERIAL_TRANSACTIONS	WF_DEFERRED
FND_APPL_SESSIONS	MTL_ONHAND_QUANTITIES_DETAIL	WF_DEFERRED_TABLE_M
FND_COLUMNS	MTL_RESERVATIONS	WF_ITEMS
FND_COMPILED_MENU_FUNCTIONS	MTL_SALES_ORDERS	WF_ITEM_ACTIVITY_STATUSES
FND_CONCURRENT_PROCESSES	MTL_SERIAL_NUMBERS	WF_ITEM_ACTIVITY_STATUSES_H
FND_CONCURRENT_REQUESTS	MTL_SUPPLY	WF_ITEM_ATTRIBUTE_VALUES

FND_CONC_PP_ACTIONS	MTL_SYSTEM_ITEMS_TL	WF_LOCAL_ROLES
FND_CRM_HISTORY	MTL_TXN_REQUEST_HEADERS	WF_LOCAL_USER_ROLES
FND_DESCR_FLEX_COLUMN_USAGES	OE_ORDER_HEADERS_ALL	WF_MESSAGES_TL
FND_DESCR_FLEX_COL_USAGE_TL	OE_SALES_CREDITS	WF_MESSAGE_ATTRIBUTES
FND_DOCUMENTS_LONG_RAW	OKC_K_HEADERS_ALL_B	WF_MESSAGE_ATTRIBUTES_TL
FND_DOC_SEQUENCE_AUDIT	OKC_K_LINES_B	WF_NOTIFICATIONS
FND_ENV_CONTEXT	ONT_DBI_CHANGE_LOG	WF_NOTIFICATION_ATTRIBUTES
FND_FLEX_VALUES	PAY_ACTION_INFORMATION	WF_PROCESS_ACTIVITIES
FND_FLEX_VALUES_TL	PAY_ACTION_INTERLOCKS	WF_USER_ROLE_ASSIGNMENTS
FND_FORM_FUNCTIONS	PAY_ASSIGNMENT_ACTIONS	WIP_ENTITIES
FND_FORM_FUNCTIONS_TL	PAY_BALANCE_FEEDS_F	WSH_DD_TXNS
FND_LOBS	PAY_DEFINED_BALANCES	WSH_DELIVERY_ASSIGNMENTS
FND_LOGINS	PAY_ELEMENT_ENTRIES_F	WSH_DELIVERY_DETAILS
FND_LOOKUP_VALUES	PAY_ELEMENT_ENTRY_VALUES_F	WSH_DELIVERY_LEGS
FND_MENU_ENTRIES	PAY_EXTERNAL_ACCOUNTS	WSH_DOCUMENT_INSTANCES
FND_MENU_ENTRIES_TL	PAY_LATEST_BALANCES	WSH_EXCEPTIONS
FND_NEW_MESSAGES	PAY_MESSAGE_LINES	WSH_FREIGHT_COSTS
FND_PROFILE_OPTION_VALUES	PAY_PRE_PAYMENTS	WSH_LOCATIONS
FND_STATS_HIST	PAY_PROCESS_EVENTS	WSH_NEW_DELIVERIES
FND_TABLES	PAY_RECORDED_REQUESTS	WSH_PICKING_BATCHES
FND_USER	PAY_RUN_BALANCES	WSH_REGION_LOCATIONS
GL_BC_PACKETS	PAY_RUN_RESULTS	WSH_TRIPS
GL_CODE_COMBINATIONS	PAY_RUN_RESULT_VALUES	WSH_TRIP_STOPS
GL_INTERFACE	PAY_USER_COLUMN_INSTANCES_F	XLA_AE_HEADERS
GL_JE_BATCHES	PER_ALL_ASSIGNMENTS_F	XLA_AE_LINES
GL_JE_HEADERS	PER_ALL_PEOPLE_F	XLA_AE_SEGMENT_VALUES
GL_JE_LINES	PER_PERSON_TYPE_USAGES_F	XLA_DISTRIBUTION_LINKS
GL_PERIOD_STATUSES	PER_TIME_PERIODS	XLA_EVENTS
HIST_HEAD$	PJM_UNIT_NUMBERS	XLA_TRANSACTION_ENTITIES
HXC_TIME_ATTRIBUTES	PO_ACTION_HISTORY	XNP_MSGS
HXC_TIME_ATTRIBUTE_USAGES	PO_APPROVAL_LIST_HEADERS	ZX_PARTY_TAX_PROFILE
HXC_TIME_BUILDING_BLOCKS	PO_DISTRIBUTIONS_ALL	ZX_REC_NREC_DIST
HZ_CONTACT_POINTS	PO_DISTRIBUTIONS_ARCHIVE_ALL	
HZ_CUSTOMER_PROFILES	PO_HEADERS_ALL	

Appendix C: Tables Compressed on the Oracle Production Database

Table 9 lists the tables compressed in Oracle's internal production database (GSI); most of the corresponding indexes were also compressed.

TABLE 9. LIST OF COMPRESSED TABLES IN GSI DATABASE

AP_AE_HEADERS_ALL	HZ_CUST_ACCOUNTS	OKI_SRM_006_MV
AP_AE_LINES_ALL	HZ_CUST_ACCOUNT_ROLES	OKS_BILL_CONT_LINES
AP_CHECKS_ALL	HZ_CUST_ACCT_RELATE_ALL	OKS_BILL_TXN_LINES
AP_CREDIT_CARD_TRXNS_ALL	HZ_MERGE_ENTITY_ATTRIBUTES	OKS_COVERAGE_TIMES
AP_EXPENSE_REPORT_HEADERS_ALL	HZ_MERGE_PARTY_HISTORY	OKS_COVERAGE_TIMEZONES
AP_EXPENSE_REPORT_LINES_ALL	HZ_MERGE_PARTY_LOG	OKS_K_HEADERS_B
AP_EXP_REPORT_DISTS_ALL	HZ_ORGANIZATION_PROFILES	OKS_K_LINES_B
AP_HOLDS_ALL	HZ_ORG_CONTACTS	OKS_K_LINES_TL
AP_INVOICES_ALL	HZ_PARAM_TAB	OKS_LEVEL_ELEMENTS
AP_INVOICE_DISTRIBUTIONS_ALL	HZ_PARTIES	OKS_REPROCESSING
AP_INVOICE_DISTS_ARCH	HZ_PARTY_SITE_USES	OTA_ACTIVITY_VERSIONS_TL
AP_INVOICE_LINES_ALL	HZ_PERSON_PROFILES	OTA_DELEGATE_BOOKINGS
AP_INVOICE_PAYMENTS_ALL	HZ_RELATIONSHIPS	OTA_EVENTS
AP_PAYMENT_HISTORY_ALL	HZ_STAGED_PARTIES	OTA_EVENTS_TL
AP_PAYMENT_HIST_DISTS	IBY_CREDITCARD	OTA_OFFERINGS_TL
AP_PAYMENT_SCHEDULES_ALL	IBY_EXT_BANK_ACCOUNTS	OTA_PRICE_LIST_ENTRIES
AR_CASH_RECEIPTS_ALL	IBY_TRXN_DOCUMENTS	OTA_RESOURCE_BOOKINGS
AR_CASH_RECEIPT_HISTORY_ALL	ICX_CAT_ITEMS_CTX_HDRS_TLP	PASA_EXPEND_EMEA2
AR_DISTRIBUTIONS_ALL	IEX_DELINQUENCIES_ALL	PAYROLL_DEDUCTIONS_NEW
AR_PAYMENT_SCHEDULES_ALL	IEX_SCORE_HISTORIES	PAY_ACTION_INFORMATION
AR_RECEIPTS_REP_ITF	INV_TOTAL_TAX_NULL	PAY_ASSIGNMENT_ACTIONS
AR_RECEIVABLE_APPLICATIONS_ALL	IPM_DOCUMENT_METRICS	PAY_ASSIGNMENT_LATEST_BALANCES
ASO_QUOTE_HEADERS_ALL	IPM_LOBS	PAY_COSTS
ASO_QUOTE_LINE_DETAILS	IPM_OBJECTS	PAY_ELEMENT_ENTRIES_F
AS_ACCESSES_ALL_ALL	IPM_OBJECT_H	PAY_ELEMENT_ENTRY_VALUES_A
AS_OPP_WORKSHEET_LINES	IPM_OBJECT_METADATA	PAY_ELEMENT_ENTRY_VALUES_F
AS_SALES_CREDITS_DENORM	JDR_ATTRIBUTES	PAY_PAYROLL_ACTIONS
AS_SALES_LEADS	JDR_COMPONENTS	PAY_PERSON_LATEST_BALANCES
AX_SLE_LINES	JTF_NOTES_B	PAY_PROCESS_EVENTS
BEN_EXT_CHG_EVT_LOG	JTF_NOTE_CONTEXTS	PAY_RUN_BALANCES
BEN_EXT_RSLT_ERR	JTF_RS_GROUPS_TL	PAY_RUN_RESULTS

BEN_PERSON_ACTIONS	JTF_RS_REP_MANAGERS	PAY_RUN_RESULT_VALUES
CN_COMMISSION_HEADERS_ALL	JTF_RS_RESOURCE_EXTNS	PA_ACTION_SET_LINE_AUD
CN_COMMISSION_LINES_ALL	JTF_RS_RESOURCE_EXTNS_TL	PA_COST_DISTRIBUTION_LINES_ALL
CN_COMM_LINES_API_ALL	JTF_RS_ROLE_RELATIONS	PA_CUST_REV_DIST_LINES_ALL
CN_COMM_LINES_API_ALL_ARCH	JTF_TASKS_TL	PA_DRAFT_INVOICE_ITEMS
CN_TRX_SALES_LINES_ALL	MIG_BALANCES	PA_EI_DENORM
CRM_ACCTMPG_RESULTS_OD	MIG_CODE_COMBINATIONS	PA_EVENTS
CSI_IP_ACCOUNTS	MISAP_EXPENSES_AUDIT	PA_EXPENDITURES_ALL
CSI_ITEM_INSTANCES	MISAR_BUCKET_AGING	PA_EXPENDITURE_COMMENTS
CSI_I_PARTIES	MISCN_COMM_LINES_API_ALL_ARCH	PA_EXPENDITURE_ITEMS_ALL
CSI_I_PRICING_ATTRIBS	MISCN_MIS_COMM_LN_API_ALL_ARCH	PA_FORECAST_ITEMS
CSI_SYSTEMS_B	MISETD_TRANSACTIONS_FACT	PA_FORECAST_ITEM_DETAILS
CSI_SYSTEMS_TL	MISETD_TRANS_FACT_AGG	PA_PROJECTS_ALL
CST_REVENUE_RECOGNITION_LINES	MISGL_IR_REPORT_REQUESTS	PA_PROJECT_ASSIGNMENTS
CZ_LOCALIZED_TEXTS	MIS_CSI_IB_CDR	PA_PROJECT_CLASSES
CZ_PS_NODES	MIS_OKC_CDR_LINES_ALL	PA_PROJECT_PARTIES
ECX_DOCLOGS	MIS_OTA_LETTER_REQUESTS	PA_RESOURCES_DENORM
FII_AP_INV_B	MIS_OTA_OLN_MEMBERSHIPS	PA_SCHEDULES
FII_AR_REVENUE_B	MIS_SUPPORT_DETAILS_ALL	PA_TASKS
FII_GL_JE_SUMMARY_B	MIS_WSH_DHL_DELIVERIES	PA_TXN_INTERFACE_AUDIT_ALL
FND_ATTACHED_DOCUMENTS	MLOG$_ENI_OLTP_ITEM_STAR	PER_ALL_ASSIGNMENTS_F
FND_CONCURRENT_REQUESTS_ARCH	MLOG$_OKI_DBI_CLE_B	PER_ALL_PEOPLE_F
FND_DOCUMENTS	MTL_ITEM_REVISIONS_TL	PER_COMPETENCE_ELEMENTS
FND_DOCUMENTS_TL	MTL_SYSTEM_ITEMS_TL	PER_OBJECTIVES
FND_FLEX_EXCLUDE_RULE_LINES	MTL_TRANSACTION_ACCOUNTS	PER_PERSON_LIST
FND_FLEX_VALUES_TL	OE_AGREEMENTS_B	PER_PERSON_TYPE_USAGES_F
FND_GRANTS	OE_AGREEMENTS_TL	PER_PHONES
FND_LOGINS	OE_ORDER_PRICE_ATTRIBS	PJI_FP_XBS_ACCUM_F
FND_LOOKUP_VALUES	OE_PRICE_ADJUSTMENTS	PO_DISTRIBUTIONS_ALL
FND_MENU_ENTRIES_TL	OE_SALES_CREDITS	PO_HEADERS_ALL
FND_NEW_MESSAGES	OFS_BOM_OPTIONS_PRICES	PO_LINES_ALL
FND_PROFILE_OPTION_VALUES	OFS_FEE_ADJUSTMENTS	PO_LINE_LOCATIONS_ALL
GL_ACCOUNT_HIERARCHIES	OFS_MIGRATION_LINES_ALL	PO_REQUISITION_HEADERS_ALL
GL_BALANCES	OFS_ORDERS_ALL	PO_REQUISITION_LINES_ALL
GL_BUDGET_ASSIGNMENTS	OFS_ORDER_MESSAGES	PO_REQ_DISTRIBUTIONS_ALL
GL_CODE_COMBINATIONS	OFS_ORDER_ROLES_ALL	PV_ENTITY_RULES_APPLIED
GL_CONSOLIDATION_AUDIT	OKC_ARTICLE_VERSIONS	QP_LIST_LINES
GL_IMPORT_REFERENCES	OKC_CONTACTS	QP_PRICING_ATTRIBUTES

GL_INTERFACE_HISTORY	OKC_K_ARTICLES_B	RA_CUSTOMER_MERGES
GL_JE_HEADERS	OKC_K_GRPINGS	RA_CUSTOMER_TRX_ALL
GL_JE_LINES	OKC_K_HEADERS_ALL_B	RA_CUSTOMER_TRX_LINES_ALL
GPI_ERROR_LOG_TL	OKC_K_HISTORY_B	RA_CUST_TRX_LINE_GL_DIST_ALL
HRI_ADM_MTHD_RANGES	OKC_K_HISTORY_TL	RA_CUST_TRX_LINE_SALESREPS_ALL
HRI_CS_SUPH	OKC_K_ITEMS	RA_INTERFACE_SALESCREDITS_ALL
HR_ALL_ORGANIZATION_UNITS_TL	OKC_K_LINES_B	RA_INTERFACE_SALESCREDITS_B
HR_ORGANIZATION_INFORMATION	OKC_K_LINES_TL	RCV_SHIPMENT_HEADERS
HR_TRANSACTION_SNAPSHOT	OKC_K_PARTY_ROLES_B	WF_DEFERRED
HXC_APP_PERIOD_SUMMARY	OKC_K_PARTY_ROLES_TL	WF_ERROR
HXC_LATEST_DETAILS	OKC_K_VERS_NUMBERS	WF_ITEMS
HXC_TIMECARD_SUMMARY	OKC_OPERATION_LINES	WF_ITEM_ACTIVITY_STATUSES
HXC_TIME_ATTRIBUTES	OKC_PRICE_ATT_VALUES	WF_ITEM_ACTIVITY_STATUSES_H
HXC_TIME_ATTRIBUTE_USAGES	OKC_RULES_B	WF_ITEM_ATTRIBUTE_VALUES
HXC_TIME_BUILDING_BLOCKS	OKC_RULES_TL	WF_NOTIFICATIONS
HXC_TRANSACTIONS	OKC_RULE_GROUPS_B	WF_NOTIFICATION_ATTRIBUTES
HXC_TRANSACTION_DETAILS	OKC_TIMEVALUES_B	WF_NOTIFICATION_OUT
HXT_DET_HOURS_WORKED_F	OKC_TIMEVALUES_TL	WSH_DELIVERY_DETAILS
HXT_SUM_HOURS_WORKED_F	OKI_COV_PRD_LINES	XLA_AE_SEGMENT_VALUES
HZ_CONTACT_POINTS	OKI_DBI_CLE_B	ZX_REC_NREC_DIST
HZ_CUSTOMER_PROFILES	OKI_SRM_005_MV	

Appendix D: Recommended Additional Reading

This appendix lists the associated papers:

- 11g Advanced Compression White Paper
  http://www.oracle.com/technology/products/database/oracle11g/pdf/advanced-compression-whitepaper.pdf
- Using Database Partitioning with Oracle E-Business Suite My Oracle Support Knowledge Document 554539.1
- Database Initialization Parameters for Oracle Applications Release 12: My Oracle Support Knowledge Document 396009.1
- Chapter 5, "Using Partitioning for Information Lifecycle Management" of Oracle® Database VLDB and Partitioning Guide 11g Release 1 for details on Information Lifecycle Management

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

Oracle E-Business Suite Release 12.1 with
Oracle Database 11g Advanced Compression
April 2010
Authors: Uday Moogala, Lester Gutierrez,
Andy Tremayne

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