

Oracle E-Business Database Shrinking and Compression Strategy

By Huw Price



Grid-Tools

The power of test data

Specialists in data creation, data masking and test data management

Database shrinking

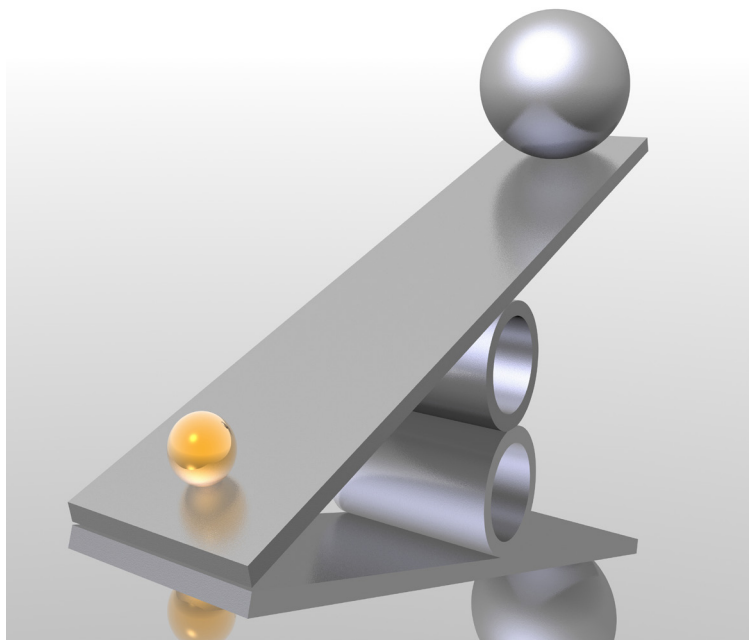
Many sites struggle with full sized copies of production databases. These databases are used for testing of new releases, reporting, development, training and testing. The management of these databases is complex and error prone and the cost of the underlying disk expensive. The ability to quickly shrink a database in situ is an attractive option as it allows database code to remain as is and requires very little application management.

An Oracle E-Business suite database is a complex structure. It contains many internal Oracle objects which cannot be easily extracted and re-assembled in a subset database. This paper discusses the method to shrink a database in place and also to implement Oracle compression as part of the database reorganization.

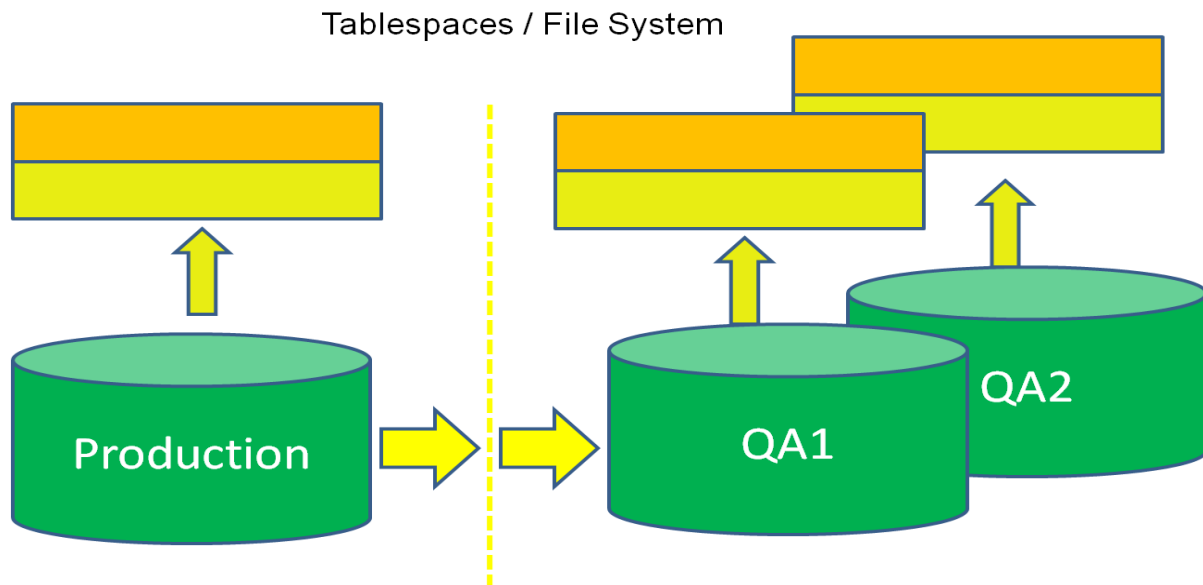
The advantages of in-situ shrinking are:

- Overall Reduced disk space
- Faster run times in testing and development databases
- Removing redundant data tables built up over time
- Specific databases can be created for different purposes (i.e. Training and Upgrade tests)

The technology used to shrink the database was Datamaker™'s Data Subset™ tool in conjunction with Oracle 's database compression along with operating system compression. The paper is split into two sections; the first outlines the use of the tool to shrink an E-business module in situ and the second is a case study on Oracle's 10g compression (see page 4).



Full sized Production Copies to Development



Mirroring / Replication / Transport

Many sites use cloning technology to migrate copies of production for testing and development.

E-Business Database Shrinking

Datamaker™ is a powerful test and development data tool set. One of the options available is a powerful script generator, which allows users to generate all the DBA scripts to shrink the database and reassemble any invalid objects. The subsetting tool comes pre-populated with standard data subsetting definitions. These definitions allow the most common modules and large tables to be subsetted out of the box. Supported modules include:

General Ledger

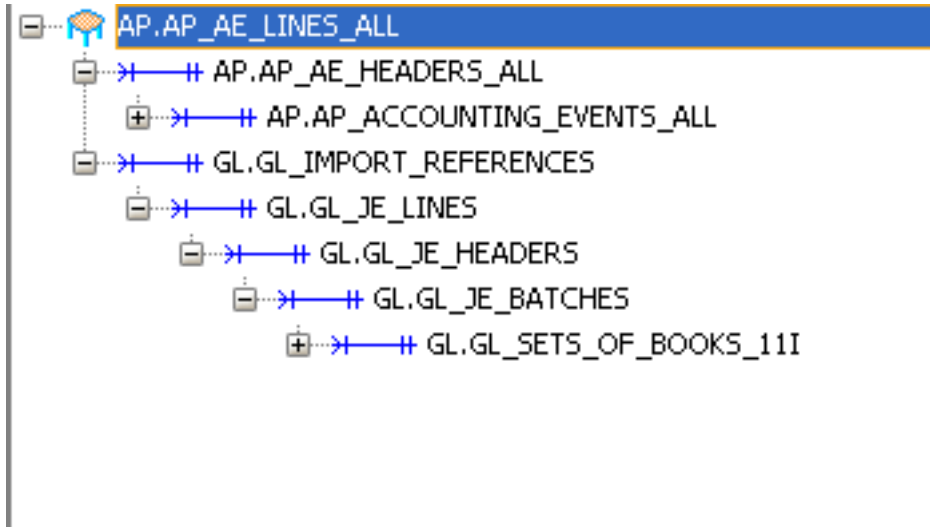
Account Payable

Accounts receivable

Contact Grid Tools for details of other modules

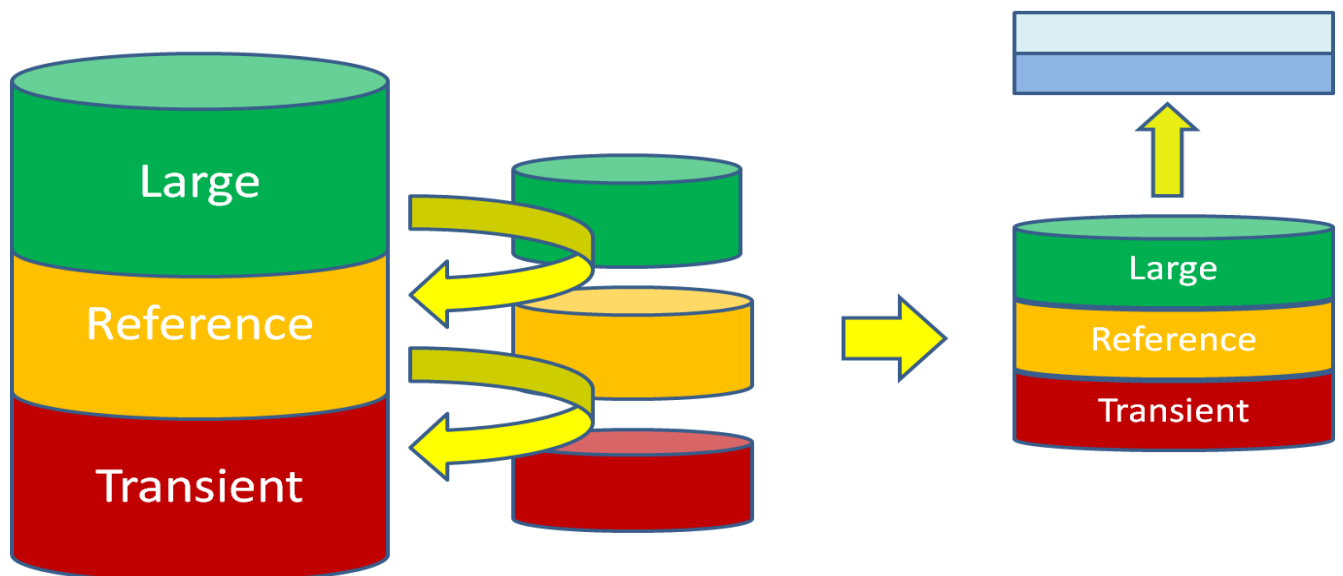
The user can also add-in their own tables to standard extracts, add-in or omit entire tables, or build their own data shrink subset definitions.

Example Modules Shrink Set



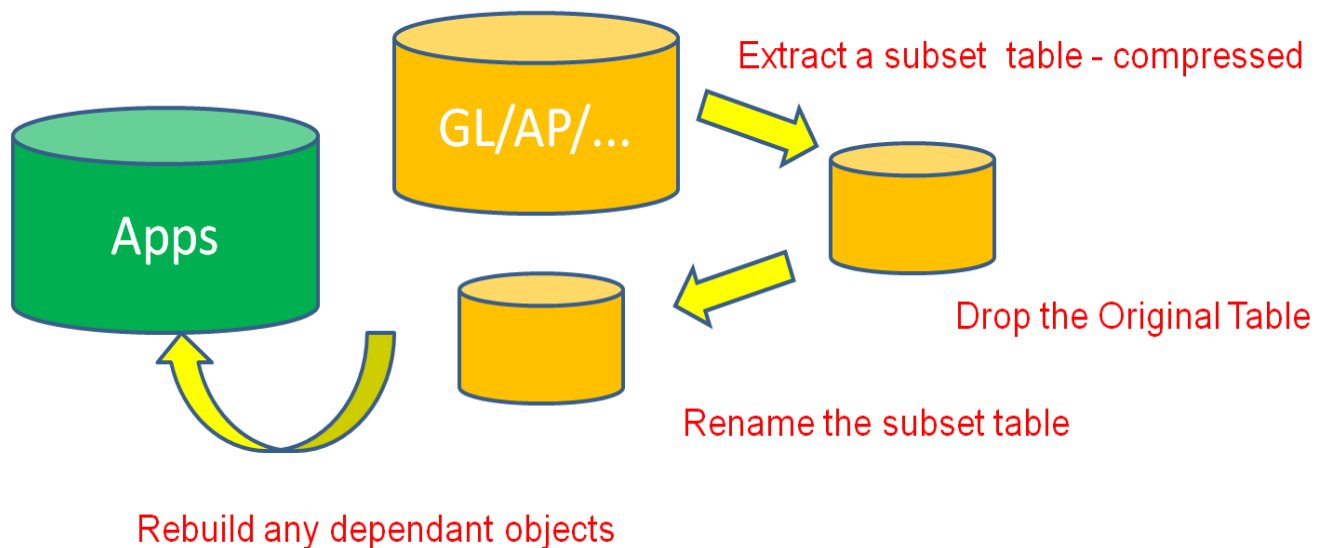
Each E-Business module has a standard set of shrinking criteria included. Users can add-in their own tables and adjust any criteria. For example, this includes sets of books, transaction dates, organization id, etc.

Subset Tables in situ and compress



The generated scripts run in parallel

Database Shrink steps



The generated scripts build all of the required UNIX or Windows scripts to shrink the database in situ. Many steps are run in parallel and contain comprehensive error checks.

The major steps are:

- Enter selection criteria
- Build temporary compressed tables
- Drop the original tables
- Rename the temporary tables
- Rebuild and reassemble any dependant objects
- Recompile any APPS objects

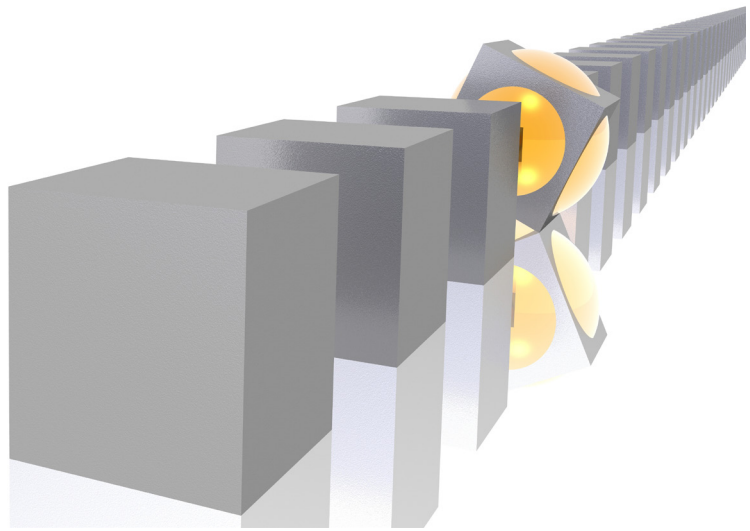
New table spaces can be specified and new space requirements are automatically calculated.

Oracle Compression

As part of the database shrink process, tables and appropriate index are redefined using compression. The following section outlines in more detail a case study on Oracle's compression technology.

Oracle versions 10g and beyond introduced a table and index compression feature.

As advertised by Oracle, compression can bring significant disk savings at three-and-a-half to one, and tangible performance improvements. This paper briefly outlines a summary of this technology and the results of tests performed by Grid Tools. Some research has been published on the web (see: Compression References). Oracle has also provided a lot of material on this subject. The tests were run to verify the research and simulated table insert and updates as utilized by typical transaction systems.



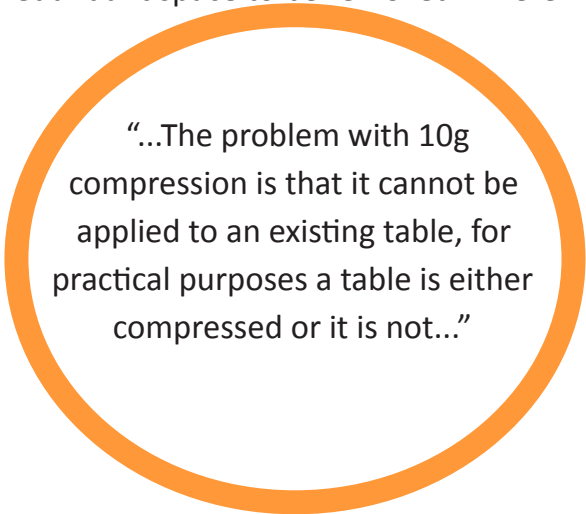
The tests also included a look at Oracle compression in conjunction with operating system compression, in this case Windows disk compression, however more work would have to be done with other operating systems to verify these results.

The results of the test do seem to verify that Compression is a valuable technology and could bring disk savings and performance improvements immediately. It has the added advantage of being able to be introduced in a phased way, (i.e. table by table reducing the need for excessive swing space). There are however testing implications, especially in the area of performance.

The Technology

10g Compression

A new verb COMPRESS has been added to the CREATE TABLE & CREATE INDEX statements. This allows these objects to contain compressed data. Each row of a table basically takes up less space. As very wide tables contain a lot of character columns, there is scope for a large amount of redundant space to be removed. There is a CPU overhead as a row is INSERTED, as it has to pass through the compress algorithm. Likewise, when a row is updated it has to be decompressed and then compressed again. There is, however, a significant benefit whereby, when a block of data is retrieved, the number of rows in the block increases, reducing the number of blocks that need to be retrieved. This is where the performance benefit comes from.



“...The problem with 10g compression is that it cannot be applied to an existing table, for practical purposes a table is either compressed or it is not...”

There is significant detail on how this works in a paper produced by Oracle:

<http://www.vldb.org/conf/2003/papers/S28P01.pdf>

The problem with 10g compression is that it cannot be applied to an existing table. For practical purposes, a table is either compressed or it is not. It is possible to compress a table in situ but the table has to be locked for the period or a PL/SQL package has to be run. The problem is not quite so bad for indexes, as these are rebuilt from time to time and can be rebuilt with the compress option enabled.

Compression occurs when data is being inserted, updated, bulk inserted, or bulk loaded into a compressed table. These operations include:

Direct Path SQL *Loader

CREATE TABLE and AS SELECT statements

Parallel INSERT (or serial INSERT with an APPEND hint) statements

Single-row or array inserts
Single-row or array updates

A strategy which was used effectively during the test was Parallel inserts. A new table was created with no indexes and was populated using parallel inserts from the existing table. The existing table was dropped, the new table renamed to the old and the indexes rebuilt.

The tests run by Grid-Tools simulated tables and indexes used in a standard transaction system and were run on version 10.2.0.1.0 of Oracle.

11g Compression

Most sites are not planning to upgrade to 11g for some time, however, it may be worth bearing in mind some of the new technology available in 11g. See Oracle's paper: <http://www.oracle.com/technology/deploy/performance/pdf/s291861.pdf>

The main feature of interest is the ability for a table to be changed to compressed without having to compress all of the data in the table. The compressed and uncompressed data can co-exist alongside one another and non compressed data can be migrated to compressed in the background. There are also some improvements in the compression algorithms themselves, which may bring some more disk space savings.

Tests and results

Three versions of a complex table ORDER were created:

1. Uncompressed
2. Compressed Using Oracle compression
3. Compressed Using Oracle compression and also using operating system compression

Each version of the table was created in its own table space. The associated 10 ORDER indexes were created in similar fashion and in their own table spaces. The total index size was designed to be similar in size to the data size.

There were 3 million rows created in each table by Grid-Tools Datamaker™. The data was randomly created to simulate live data with random length text and a spread of nulls and populated columns. The rows were then doubled in size by applying updates to data columns and indexes columns. This was done to fragment the data rows across multiple blocks.

Space savings

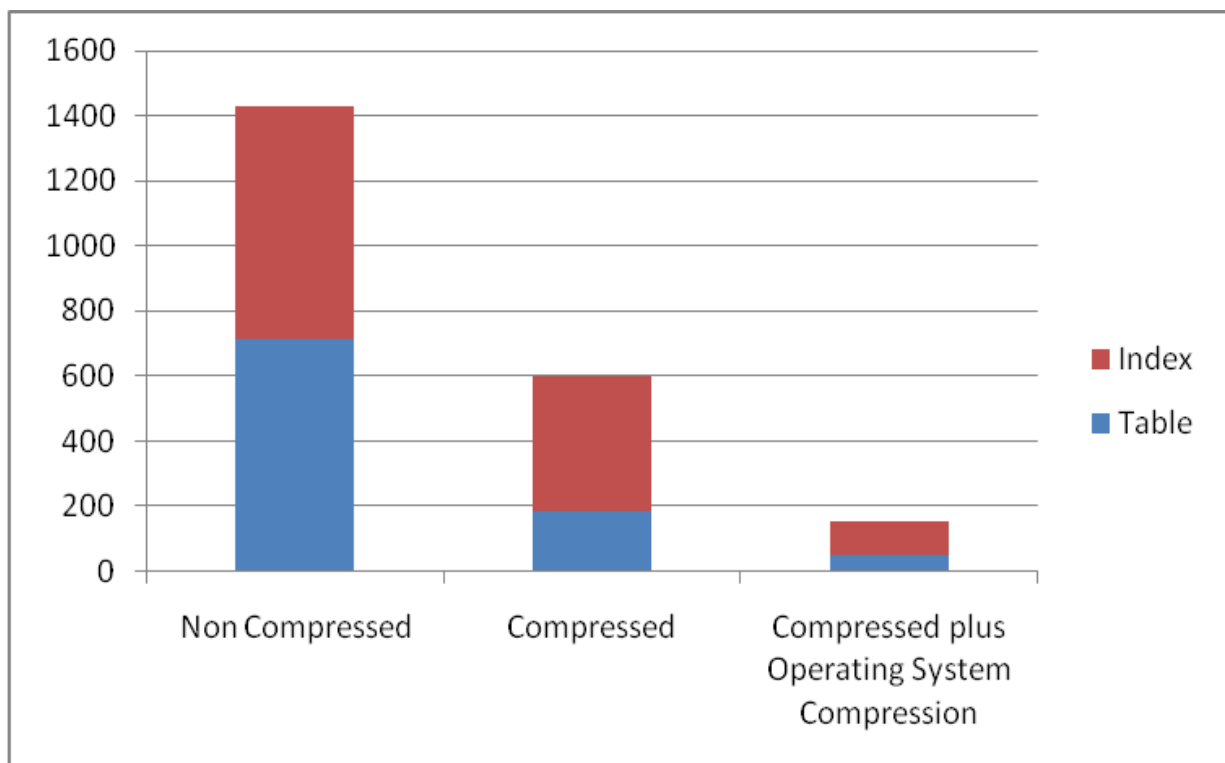
Rows

Non Compressed: Average rows size 261 Bytes – 31 Blocks per row

Compressed: Average rows size 66 Bytes – 123 Blocks per row

A 4 to 1 improvement.

Table and Index Space in Megabytes



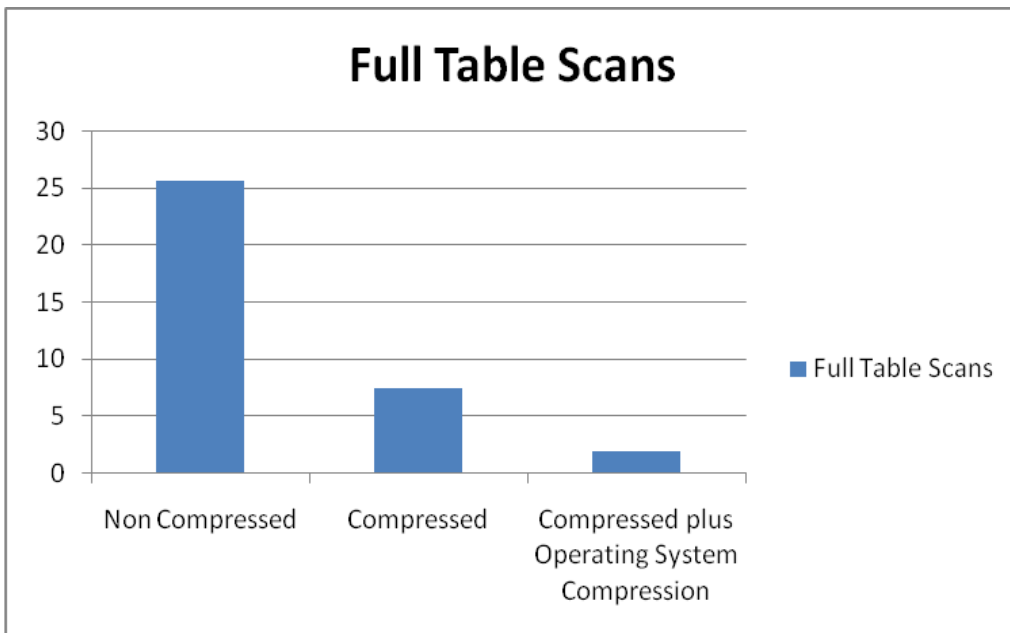
A 2.4 to 1 Improvement between Oracle Compressed and Non Compressed tables.

Performance savings

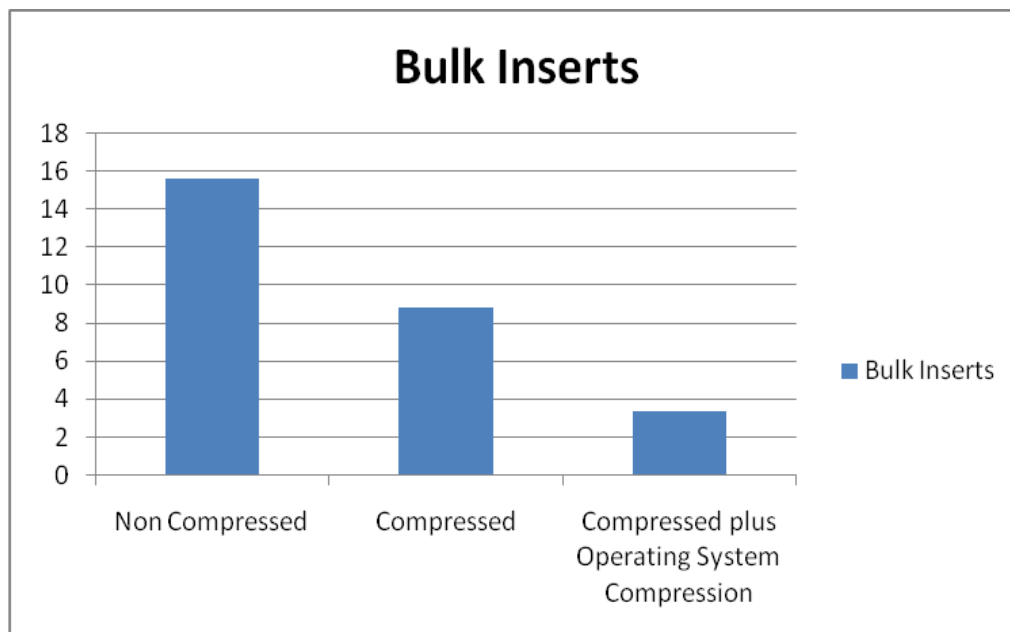
A number of test were performed, including a full table scan select, a number of bulk inserts and large updates.

Full Table Scan

Number of seconds for a full table scan query:



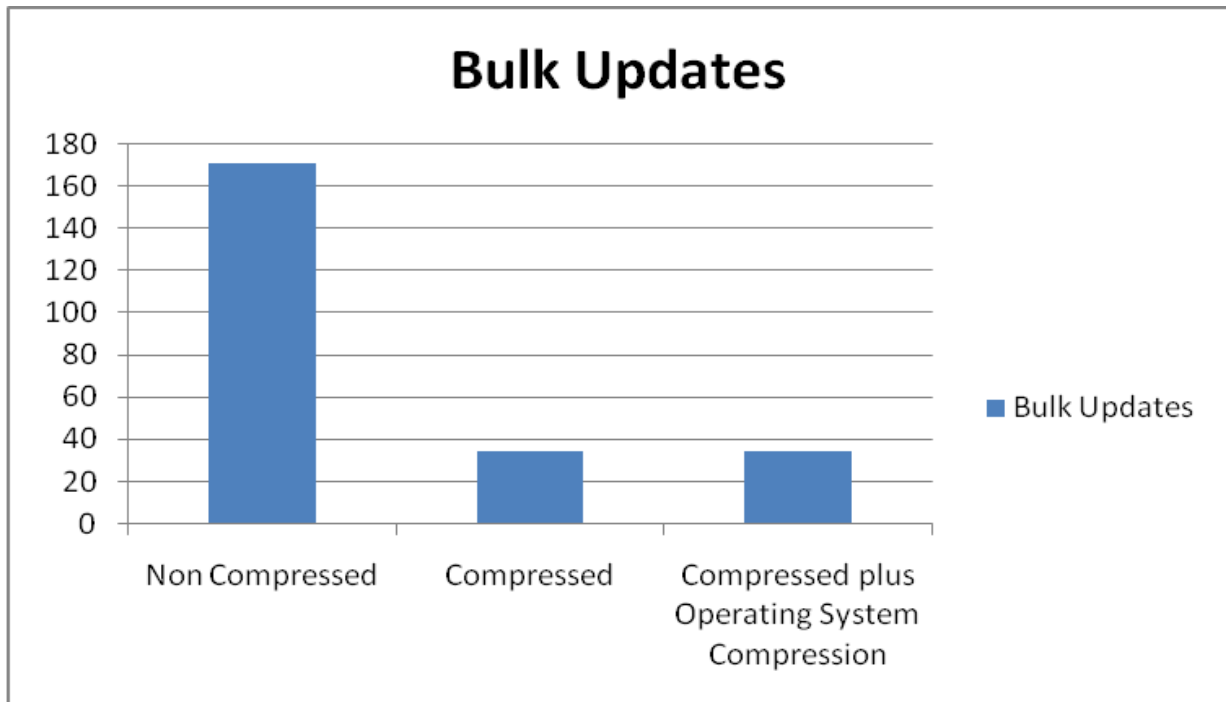
A 3.5 to 1 Improvement between Oracle Compressed and Non Compressed tables.



A 1.8 to 1 Improvement between Oracle Compressed and Non Compressed tables.

Bulk Updates

Time in seconds to update 10,000 rows doubling the data sizes to simulate block defragmentation:



A 5 to 1 improvement between Oracle Compressed and non compressed tables.

This was the most surprising, as updates require an uncompress and recompress. It is most likely that the speed increase was due to the improved selection speed and reduced I/O transfer. The extra CPU was probably insignificant compared to these improvements.

Summary and Recommendations

Based on these test results, which seem to verify and in some cases out-perform Oracle's published results, the use of compression could bring immediate disk savings and potential performance improvements. In conjunction with hardware compression, some queries were a much faster 14:1, however a more realistic 3:1 would be a reasonable expectation. The disk savings were significant and worked out at a 2.4:1 saving without hardware compression.

Partitioning Strategy

Some of the research suggests that compressing your older data (older partitions) and leaving later partitions uncompressed is a good strategy, as the newer data may well be updated and cause defragmentation. This strategy, however, would bring added complication to the implementation of partitioning, as it would involve re-compressing older partitions as they age.

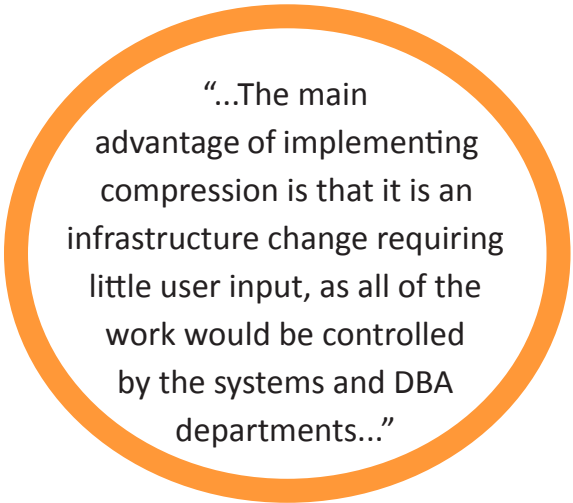
Next Steps

Before sites implement compression it is suggested that they perform some pilots and some performance tests, these could include:

- Migrating several medium and large sized tables to compressed versions while utilizing the parallel load facilities
- Rebuilding a number of indexes in compressed format
- Using the opportunity to reorganize some of the table spaces while migrating the tables
- Gathering bench mark statistics on before and after tables, capturing plans, run times, cpu usages etc.
- Gathering estimates on how long it takes to migrate the larger tables to compressed versions.

Compression Advantages

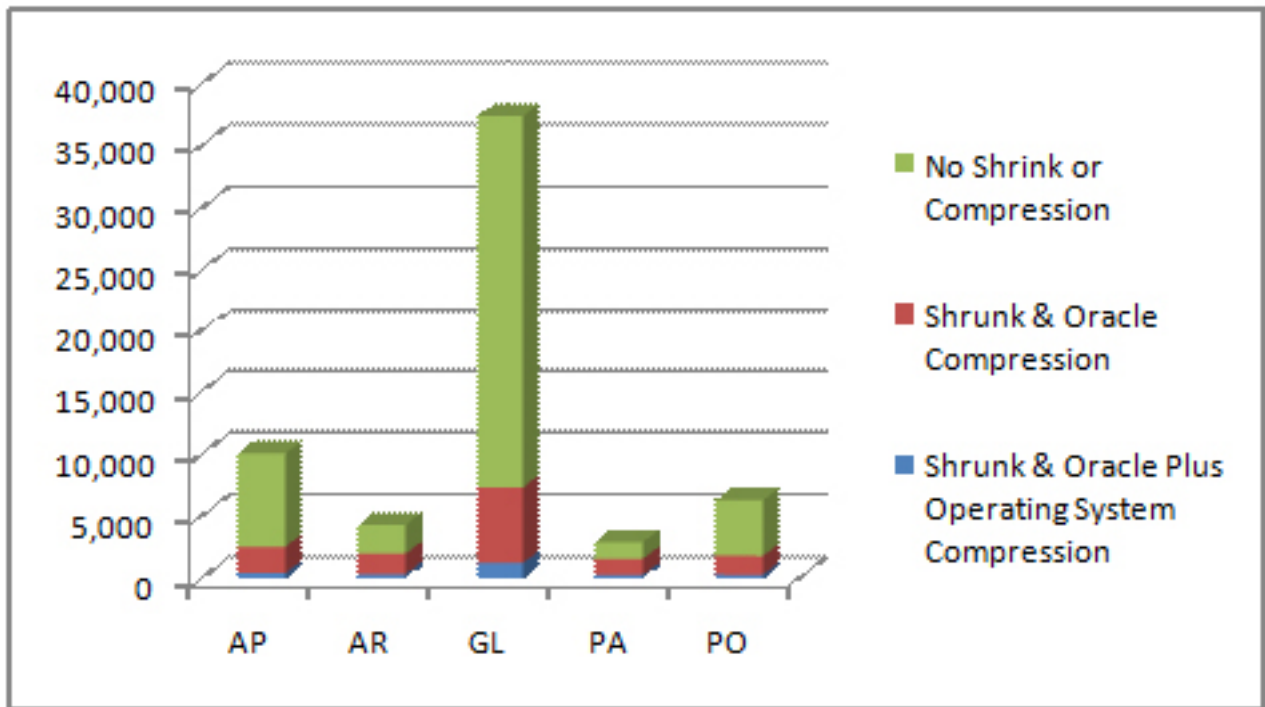
The main advantage of implementing compression is that it is an infrastructure change requiring little user input, as all of the work would be controlled by the systems and DBA departments. Data could be migrated with little swing space as portions of the entire system are migrated over in phases. The savings in disk space look significant and we have realized measurable performance improvements.



“...The main advantage of implementing compression is that it is an infrastructure change requiring little user input, as all of the work would be controlled by the systems and DBA departments...”

E-Business Shrinking and compression

Using database shrinking combined with compression can significantly reduce your data footprint.



Sample database shrinking and compression for an e-business application, assuming a 20% reduction using organization ID, pivot dates or sets of books (size in meg.)

Compression References

Further reading can be found at:

http://www.dba-oracle.com/oracle_news/2005_9_16_data_compression_benefits.htm

<http://www.vldb.org/conf/2003/papers/S28P01.pdf>

<http://husnusensoy.wordpress.com/2008/02/01/using-oracle-table-compression/>

About Grid-Tools Ltd

Grid-Tools are specialists in test data creation, data masking and test data management. Their experienced personnel have been writing and developing solutions for large companies in both the private and public sectors for over 30 years.

The Grid-Tools Datamaker™ suite includes a wide range of tools for test data management including such innovative products as Datamaker™, a revolutionary tool that creates and publishes quality test data from scratch whilst keeping the relationships and referential integrity of production environments. An invaluable tool for testing and development, Datamaker™ places the data into a central data repository so it can be used, inherited, manipulated and re-used across an entire organization. Voted “Most Innovative Testing Tool of 2008” by QA Guild, the functionality Datamaker™ provides has proven to be innovative, efficient and different than any other test data management product in the market. You simply will not find another tool like it!

About Huw Price

With over a 30 year career, Huw Price has been the lead technical architect for several US and European software companies. Specializing in test automation tools, he has launched numerous innovative products which have re-cast the testing model used in the software industry. Huw has provided high-level architecture design support to multinational banks, major utility vendors and health care providers.

A strong believer in balancing pragmatism with a visionary approach, he has been able to rapidly bring new products to market while maintaining strong quality control. Huw’s newest venture, Grid-Tools, has quickly redefined how large organizations need to approach their test data management strategy.

Grid-Tools contact details

Grid-Tools Headquarters:
11 Oasis Business Park
Eynsham, Oxford
England
OX29 4TP

UK: +44 01865 884600
USA: 1-866-563-3120

Email: sales@grid-tools.com

© Grid-Tools Ltd 2010

www.grid-tools.com

info@grid-tools.com

UK: +44 (0) 1865 884 600

USA: 1 866 563 3120