

May 2006

IBM Information Management software



Compression in DB2 Viper

*IBM Software Group
Toronto Laboratory*

Contents

1. Introduction
2. What is row compression
3. How compression works and how to use it
4. Real world results
5. Competitive comparison
6. Conclusions

Introduction

As the amount of data being managed continues to rise, the cost of storage is becoming an increasing burden to CIOs and IT executives. Although the cost of disk drives as measured in dollars per gigabyte of storage continues to decline, the overall IT expense on storage infrastructure continues to climb. This is as a result of the explosive growth in storage requirements of both structured and unstructured data that needs to be managed. Looking at the cost of database storage along with the cost of backup media and disaster recovery storage, a significant amount of money can be saved if the database were to consume a substantially less space.

DB2 Viper brings to the table a significant advancement with its new compression feature. This feature allows databases to be compressed by 45% on average thus reducing storage costs and improving performance in many cases. In addition, backup images are now half the size and therefore backup/restore operations now finish in almost half the time and consume half the space. Multiply that by the number of backup images you keep and the savings grow even larger. In addition, disaster recovery sites now require almost half the amount of storage as well. Looking at the cost of enclosures, controllers, power, cooling, floor space, etc., you can quickly see that DB2 Viper Compression can save a significant amount of money for the enterprise.

What is row compression

DB2 Viper uses a dictionary based Lempel-Ziv form of compression. Repeating patterns within a table are extracted from the rows and stored once in a dictionary. These patterns are then replaced with a 12-bit symbol within the row itself. As shown in Figure 1, repeating column values can be replaced as well as strings that repeat across column boundaries.

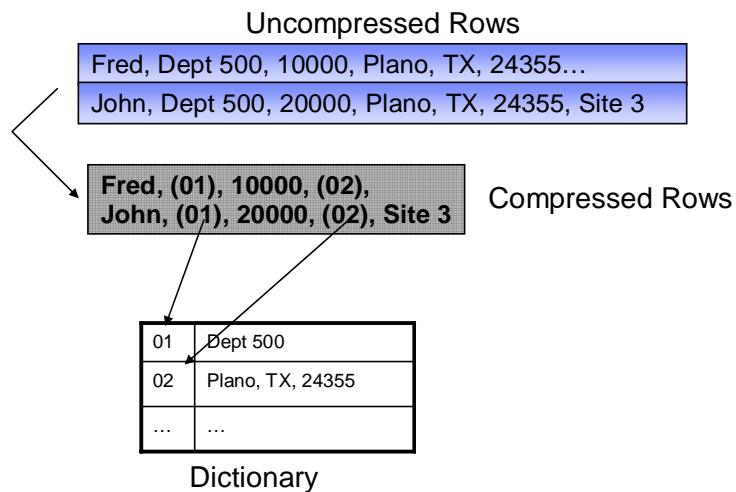


Figure 1

Highlights

The amount of compression achieved in this manner on real customer data during the DB2 Viper test drive has been reported at between 50% and 80%.

Using this form of compression, DB2 can scan the entire table, find repeating strings within the data and replace those strings with 12bit symbols. The amount of compression achieved in this manner on real customer data during the DB2 Viper test drive has been reported at between 50% and 80%. That is, the size of the table after compression is between 50% and 80% smaller than when it started. Thus the storage consumption for these tables is significantly less. In addition, DB2 keeps these data compressed in the buffer pool so that you effectively have twice the amount of memory available. That is, since rows remain compressed in the buffer pool, you can expect to see twice as many rows being stored in memory compared to non-compressed data. In one test of an SAP table with real customer data, the number of rows packed into the bufferpool was in fact four times the number of rows achieved without compression.

How compression works and how to use it

To compress a table in DB2 Viper there are two steps involved. The first is to make the table eligible for compression. Simply create the table or alter the table and specify COMPRESS YES. This tells DB2 that you want to use compression algorithms on this table.

The second step is to build a dictionary of repeating values that exist within the table. To perform this task, simply run

```
REORG TABLE table_name
```

The reorg will create the dictionary of common values (if one does not exist), and then compress those values out of the data as the table is being reorganized. Future reorgs give you the option to keep the dictionary or rebuild a new one if data patterns change. From this point onward, any insert, update, import or load of data into the table will respect the dictionary information and compress those rows as well.

DB2 also has a compression estimation utility. If you are using DB2 Viper and you want to know how much compression will benefit a given table you can use the INSPECT command. Running the following command will provide compression estimates for the given table:

```
INSPECT ROWCOMPESTIMATE TABLE NAME table_name  
RESULTS KEEP file_name
```

In a test of this feature, the compression estimate for two SAP tables was 75% for one and 74% for the second. After compressing the two tables the actual savings were 75.4% and 74.5% respectively.

After compressing the two [SAP] tables the actual savings were 75.4% and 74.5% respectively

Real world results

As part of the DB2 Viper test drive program, several customers have tested the compression capabilities. In one test, a customer had a table that consumed 180GB of disk space. Using DB2 Viper

Highlights

Using real customer data, DB2 was able to compress 6 large BW tables in the range of between 67% and 82%.

compression reduced the storage consumption down to just 42GB saving 76% of the disk space. In another case, a second customer used DB2 Viper compression on 7 large tables and found compression rates of between 55% and 72%.

In yet another example, we observed the compression rates for various SAP BW tables. SAP BW is a data warehousing environment with various FACT and ODS tables. Using real customer data, DB2 was able to compress 6 large BW tables in the range of between 67% and 82%. In one test, DB2 was able to compress a large fact table by 74% while a competitor was only able to compress the same table by 10%. There were also measurable performance advantages here as well. Not only do large table scans perform much more quickly due to fewer I/Os, but in tests of OLTP-like query workloads, DB2 performed much better on the compressed data due to the fact that the buffer pool had 4x more rows packed in memory.

Competitive comparisons

Looking at competitive data servers, some have compression while others do not. Microsoft SQL Server 2005 for example has no native data compression. As a result, customers with large amounts of data will consume large amounts of disk space and pay for the added costs associated with storage, backup, disaster recovery, etc. Oracle 10g does support compression however there is a very large distinction between Oracle and DB2 compression.

In Oracle 10g, compression is only performed at the block (also known as page) level. A block is a unit of storage ranging in size from 4k to 32k. Oracle looks at the rows on each block separately to find repeating column values. It then stores those values in the block header and replaces those column values within the rows with a key. This is much like the dictionary based compression in DB2 Viper with two large differences. Oracle is based upon repeating *column values* within a single block where as DB2 is looking for repeating *patterns* within the entire table. DB2 looks across column boundaries as well as within columns to compress out substrings if appropriate while Oracle only compressed the entire column value within the row if it repeats in other rows on the block. The big difference though is that by compressing across a larger domain of data, DB2 is able to deliver much higher levels of compression compared to Oracle. In fact, the following quote comes from one of Oracle's own research publications on compression [1].

DB2 is able to deliver much higher levels of compression compared to Oracle.

“Due to its global optimality of compression a table-wide dictionary approach can result in high compression factors.”

“Furthermore, a global dictionary may use less disk space compared to a block level approach, which potentially repeats dictionary entries across blocks.”

In fact, the only publicly available comparison of compression between DB2 and Oracle comes from this same paper. In this paper, the author uses the TPC-H database to demonstrate rates of compression with Oracle. TPC-H has not only a well documented

Highlights

DB2 delivers 3 times more compression than Oracle.

Oracle only supports compressing of data during bulk load operations

Using DB2 Viper row compression can save you between 50% and 80% disk space.

set of table definitions but it also has common data that can be loaded into these tables. Therefore both vendors are compressing virtually the same objects. In the Oracle paper they claim to deliver 38% compression on the LINEITEM table and only 18% compression on the ORDERS table. With DB2 Viper we are able to achieve 58% compression on the LINEITEM table and 60% compression on the ORDERS table. That's 3 times more compression with DB2.

In addition, Oracle only supports compressing data during bulk load operations [2]. Normal insert or update operations are not compressed in Oracle 10gR2. As previously mentioned, DB2 Viper supports compressing data via reorg, load, update, import and insert operations, thus maintaining better compression rates over time.

Conclusions

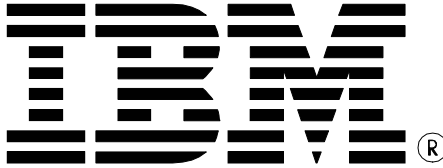
DB2 Viper brings a new level of cost control to your IT infrastructure. Using DB2 Viper row compression can save between 50% and 80% disk space on some of your largest tables. Combined with the storage savings on a disaster recovery site, backup storage savings, floor space, electrical and other infrastructure costs associated with storage, the DB2 Viper technology can result in significant savings to your business.

Add this to the performance gains achieved by scanning fewer data pages and from increased memory utilization and you can see why customers are excited about deploying the compression capabilities of DB2 Viper.

For a free test drive of DB2 Viper and to see what the compression technology can do for you, visit <http://www.ibm.com/db2/viper> today.

References

1. Data Compression in Oracle - Meikel Poess, Dmitry Potapov (Oracle Corporation, USA) - VLDB 2003 Online Proceedings <http://www.vldb.org/conf/2003/papers/S28P01.pdf>
2. Oracle Database Concepts 10g Release 2 http://download-east.oracle.com/docs/cd/B19306_01/server.102/b14220/schema.htm#sthref734



© Copyright IBM Corporation 2006
IBM Canada
8200 Warden Avenue
Markham, ON
L6G 1C7
Canada

Printed in United States of America
05-2006
All Rights Reserved.

IBM, DB2, DB2 Universal Database, OS/390, z/OS, S/390, and the ebusiness logo are trademarks of the International Business Machines Corporation in the United States, other countries or both.

UNIX and Unix-based trademarks and logos are trademarks or registered trademarks of The Open Group. Intel and Intel-based trademarks and logos are trademarks or registered trademarks of Intel Corp. Other company, product or service names may be the trademarks or service marks of others.

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

The information in this white paper is provided AS IS without warranty. Such information was obtained from publicly available sources, is current as of 05/01/2006, and is subject to change. Any performance data included in the paper was obtained in the specific operating environment and is provided as an illustration. Performance in other operating environments may vary. More specific information about the capabilities of products described should be obtained from the suppliers of those products.