Db2 Huffman Compression Usability

David Nguyen – IBM nguyend@us.ibm.com

Frances Villafuerte – IBM francesv@us.ibm.com

Tridex March Virtual Conference
03/11/2021

Agenda

- Db2 compression history
- Huffman compression characteristics
- Enabling Huffman compression
  - Two Methods
  - Object-Level Compression Enhancements
- Performance evaluation
- Utility updates
  - DSN1COMP enhancements
- Summary

Disclaimer / Trademarks

© Copyright IBM Corporation 2018. All rights reserved.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

THE INFORMATION CONTAINED IN THIS DOCUMENT HAS NOT BEEN SUBMITTED TO ANY FORMAL IBM TEST AND IS DISTRIBUTED AS IS. THE USE OF THIS INFORMATION OR THE IMPLEMENTATION OF ANY OF THESE TECHNIQUES IS A CUSTOMER RESPONSIBILITY AND DEPENDS ON THE CUSTOMER'S ABILITY TO EVALUATE AND INTEGRATE THEM INTO THE CUSTOMER'S OPERATIONAL ENVIRONMENT. WHILE IBM MAY HAVE REVIEWED EACH ITEM FOR ACCURACY IN A SPECIFIC SITUATION, THERE IS NO GUARANTEE THAT THE SAME OR SIMILAR RESULTS WILL BE OBTAINED ELSEWHERE. ANYONE ATTEMPTING TO ADAPT THESE TECHNIQUES TO THEIR OWN ENVIRONMENTS DO SO AT THEIR OWN RISK. ANY PERFORMANCE DATA CONTAINED IN THIS DOCUMENT WERE DETERMINED IN VARIOUS CONTROLLED LABORATORY ENVIRONMENTS AND ARE FOR REFERENCE PURPOSES ONLY. CUSTOMERS SHOULD NOT ADAPT THESE PERFORMANCE NUMBERS TO THEIR OWN ENVIRONMENTS AS SYSTEM PERFORMANCE STANDARDS. THE RESULTS THAT MAY BE OBTAINED IN OTHER OPERATING ENVIRONMENTS MAY VARY SIGNIFICANTLY. USERS OF THIS DOCUMENT SHOULD VERIFY THE APPLICABLE DATA FOR THEIR SPECIFIC ENVIRONMENT.

Trademarks IBM, the IBM logo, ibm.com, Db2, IBM Z and z/OS are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Db2 Compression History

	Compression technique since...	**Remarks**
Table space	V3: Lempel-Ziv and dictionary	Special processor to support CMPSC instruction
	V11: Partial Compression	Software compression
	V12 FL504: Huffman Compression	CMPSC supports Huffman encoding scheme
Index space	V9: Prefix compression	Software compression
LOB	V12: zEnterprise Data Compression (zEDC)	zEDC Express feature is required
XML	V9: Lempel-Ziv and dictionary	Special processor to support CMPSC

Why Huffman?

- Just like Lempel-Ziv, Huffman is a dictionary-driven compression and decompression algorithm
  - An encoding scheme takes advantage of the disparity between frequencies and uses less storage for the frequently occurring characters at the expense of having to use more storage for each of the rarer characters.
  - The entries in the dictionary are sorted by frequency of occurrence with the highest frequency getting a shorter bit pattern to identify the entry in the dictionary
  - It is commonly used in the industry of compression programs
- Improves compression ratio of various Db2 data by taking advantage of z14 hardware support for Huffman encoded dictionaries (entropy encoded compression)
- Provides alternate compression option to existing fixed-length compression
  - In some cases, it may require more data pages to store dictionary tree than fixed-length

Huffman Compression Prerequisites

- A Huffman dictionary is only generated when the following conditions are met:
  - Hardware support – IBM z14 or higher
  - Db2 12 function level 504 or higher
  - Object is Universal Table Space (UTS)
- If any of the above conditions are not met, Db2 will generate the classic fixed-length dictionary

- Huffman compression algorithm does not support directory object (SPT01 table space)
- Huffman does not support table spaces that are *organized by hash*.
- If data that is compressed using a Huffman-encoded dictionary is moved to a z13 or older…
  - Data can still be expanded via software
  - Performance degradation is expected

Enabling Huffman Compression – Step 1

Step 1: enable compression and specify the Huffman algorithm.

Method A:

1. Set the TS_COMPRESSION_TYPE subsystem parameter to HUFFMAN.
   - This requires Db2 12 function level 504 (V12R1M504)
   - On IBM z13? Message DSNZ020I: *The TS_COMPRESSION_TYPE subsystem parameter is set to HUFFMAN on a system without entropy encoding hardware support. Fixed-length compression algorithm is used.*
2. Specify COMPRESS YES for the table space or partition.
   - `ALTER TABLESPACE TESTDB.TSCOMP COMPRESS YES;`

Method B:

1. Specify COMPRESS YES **HUFFMAN** for the table space or partition.
   - This requires application compatibility level 509 (V12R1M509)
   - `ALTER TABLESPACE TESTDB.TSCOMP COMPRESS YES HUFFMAN;`

Enabling Huffman Compression – Step 2

Step 2: trigger compression of the table space or partition.

- INSERT or MERGE statements
  - can trigger *compression-on-insert* at a threshold
- LOAD DATA or REORG TABLESPACE utilities
  - If PH33015 is applied, DSNU244I message indicates compression algorithm used:

```
DSNU244I  -DB2A 048 18:11:39.21 DSNURWT - COMPRESSION REPORT FOR
TABLE SPACE TESTDB.TSCOMP, PARTITION 1

 58 KB WITHOUT COMPRESSION
 39 KB WITH COMPRESSION
 32 PERCENT OF THE BYTES SAVED FROM COMPRESSED DATA ROWS
 ...
 78 PAGES REQUIRED WITHOUT COMPRESSION
 83 PAGES REQUIRED WITH COMPRESSION
 -6 PERCENT OF THE DB2 DATA PAGES SAVED USING COMPRESSED DATA

    COMPRESSION TYPE IN USE:  HUFFMAN
```

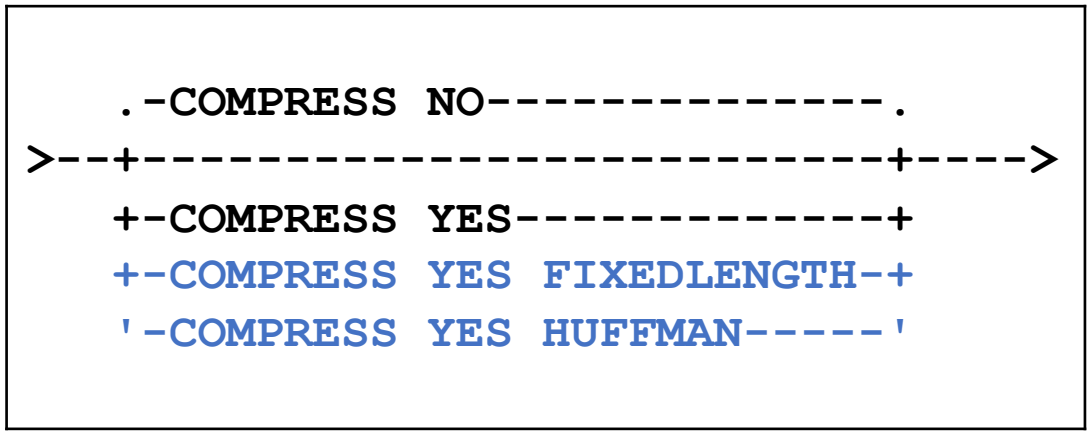
Enabling Huffman Compression – Step 3

Step 3: verify Huffman compression occurred.

- DSNU244I message from LOAD or REORG
  - after PH33015 applied
- SYSTABLEPART.COMPRESS_USED value in Db2 Catalog
  - This requires Db2 12 catalog level 509 (V12R1M509)
  - V12R1M509 activation requires PH33015.
- DSN1PRNT Utility
  - HPGZLD field on the header page describes compression type
  - before PH33015 applied

Object-Level Huffman Compression – DDL Syntax

- Db2 12 function level 509 introduces object-level Huffman compression control
  - Requires application compatibility level 509 (also known as APPLCOMPAT 509)
- FIXEDLENGTH and HUFFMAN keywords in the COMPRESS YES option in:
  - ALTER TABLESPACE
  - CREATE TABLESPACE
  - CREATE TABLE



Object-Level Huffman Compression – DDL Continued

- FIXEDLENGTH and HUFFMAN keywords in the COMPRESS YES option:
  - Change a table space, including all its existing partitions, to use Huffman compression:

```
ALTER TABLESPACE TESTDB.TSCOMP COMPRESS YES HUFFMAN;
```

  - Change a specific partition of a table space to use Huffman:

```
ALTER TABLESPACE TESTDB.TSCOMP
  ALTER PARTITION 17 COMPRESS YES HUFFMAN;
```

  - Db2 restricts unsupported combinations.
    - ✓ If `TESTDB.TSCOMP` is a non-UTS table space, Db2 returns SQLCODE -650, Reason Code 49.
    - ✓ If `TESTDB.TSCOMP` is a hash-organized table space, Db2 returns SQLCODE -650, Reason Code 51.

Object-Level Huffman Compression – DDL Continued

- FIXEDLENGTH and HUFFMAN keywords in the COMPRESS YES option:
  - Create a partition-by-growth table space that uses Huffman compression:

```
CREATE TABLESPACE TSCOMP IN TESTDB
  COMPRESS YES HUFFMAN
  MAXPARTITIONS 42;
```

  - The COMPRESS option can can be specified independently by partition. Create a partition-by-range table space that generally uses fixed-length compression, but has two partitions that use Huffman compression:

```
CREATE TABLESPACE TSCOMP IN TESTDB
  COMPRESS YES FIXEDLENGTH
  NUMPARTS 5 (
    PARTITION 2 COMPRESS YES HUFFMAN
   ,PARTITION 4 COMPRESS YES HUFFMAN);
```

Object-Level Huffman Compression – DDL Continued

- FIXEDLENGTH and HUFFMAN keywords in the COMPRESS YES option:
  - Create a table whose entire implicit table space uses the Huffman compression algorithm:

```
CREATE TABLE TBCOMP (CL1 CHAR(255))
  IN DATABASE TESTDB
  COMPRESS YES HUFFMAN;
```

Object-Level Huffman Compression – Existing Catalog Column

- Fixed-length and Huffman values in the COMPRESS catalog column
  - in the SYSTABLESPACE and SYSTABLEPART catalog tables
  - represents the *desired* compression algorithm

Y
The table space or partition is defined to use compression. If the table space is not a LOB table space, then the compression algorithm is determined by the TS_COMPRESSION_TYPE subsystem parameter.
F
The table space or partition is defined to use fixed-length compression
H
The table space or partition is defined to use Huffman compression
blank
No compression

Object-Level Huffman Compression – New Catalog Column

- Fixed-length and Huffman values in the COMPRESS_USED catalog column
  - in the SYSTABLEPART catalog table
  - represents the *used* compression algorithm

  - *COMPRESS_USED is updated after Db2 builds a compression dictionary for the partition.*

F
The table space or partition is compressed with fixed-length compression
H
The table space or partition is compressed with Huffman compression
blank
If the table space is a LOB table space and COMPRESS is Y, then zEDC hardware manages compression if available.
Otherwise, the table space or partition is not compressed.
null
The compression state is unknown, because the object was created prior to catalog level V12R1M509 and the field has not been populated.

Object-Level Huffman Compression – Additional Consideration

- If all compression criteria are met and compression occurs, the following *rare* conditions will result in the COMPRESS_USED column not being updated to reflect the compression status:
  - An insert operation holds an exclusive usage lock on an object.
  - An insert operation accesses an object started with the *UT Access* state.
  - Compression-on-insert or Utilities cannot successfully update the column.

- The following operations are not enhanced to update COMPRESS_USED to reflect the compression status:
  - A LOAD utility specifies the SHRLEVEL CHANGE option.
  - A RECOVER utility recovers data to the current state. (RECOVER to current)

Performance Evaluation

Huffman Compression Performance Consideration

- Up to 40% (avg. 20-30%) improvement compared to legacy fixed-length compression
- Wide range of variability in terms of CPU and elapsed time performance (+ / -)
  - Some cases, may see CPU and elapsed time reduction e.g., sequential processing
- The use of Huffman compression should be evaluated object by object

Example of in-house performance measurement Workload : COPY TPCH LINITEM tablespace.

	Fixed-length	Huffman	delta (%)
Compression savings (%)	54	59	9.26
Elapsed time (sec)	46.35	40.05	-13.95
CPU time (sec)	4.49	3.95	-12.03
Getpages	2641536	2332677	-11.69

Data is populated with LOAD utility – Compression savings from 54% to 59%, further 9% improvement
Measuring performance in COPY utility – see 13% elapsed time improvement
- 12% CPU time improvement
- 11% getpage reduction

Utility Enhancements

Huffman Compression Utility Support

- Support to update SYSTABLEPART.COMPRESS_USED in:
  - Db2 12 Function Level (FL) 509 introduced new catalog column
  - LOAD DATA SHRLEVEL NONE or SHRLEVEL REFERENCE
  - REORG TABLESPACE
    - ✓ DSNU244I message indicates compression algorithm used
  - RECOVER to point-in-time
  - RUNSTATS
  - REPAIR CATALOG
    - ✓ A way to update SYSTABLEPART.COMPRESS_USED column for compressed objects that exist prior to FL509

DSN1COMP Enhancements

- DSN1COMP now supports space saving estimation for either/both Huffman or Fixed-length compression algorithms.
  - New parameter COMPTYPE specifies estimation for different compression algorithm
    - ✓ Default is same as COMPTYPE(ALL) – output provides both Huffman and Fixed-length estimates
  - Requires APAR PH19242
    - ✓ Only supports input files containing non-compressed data
- Support input data file that contains compressed data
  - Requires APAR PH34808
  - Provides estimation to determine if new dictionary should be built with the same compression algorithm
  - Message describes the data compression state of input file
- Huffman compression estimation only supports Universal Table Spaces
  - The output only shows FIXED-length compression estimation

```
EXEC PGM=DSN1COMP, or EXEC PGM=DSN1COMP,PARM='COMPTYPE(ALL)' or
                                        PARM='COMPTYPE(HUFFMAN)' or
                                        PARM='COMPTYPE(FIX)'
```

Example of DSN1COMP – 1

FIXED COMPRESSED CASE when COMPTYPE(ALL) specified:

```
DSN1998I INPUT DSNAME = DB2SMS.DSNDBC.DSN03545.ZZINSRGR.I0001.A001  , VSAM
DSN1944I DSN1COMP INPUT PARAMETERS
```

```
INPUT DATA SET CONTAINS COMPRESSED DATA
USING FIXED-LENGTH COMPRESSION TYPE
INPUT DICTIONARY BUILT BY INSERT
```

```
         4,096  DICTIONARY SIZE USED
             0  FREEPAGE VALUE USED
             5  PCTFREE VALUE USED
        COMPTYPE(ALL) REQUESTED
               NO ROWLIMIT WAS REQUESTED
               ESTIMATE BASED ON DB2 LOAD METHOD
           255  MAXROWS VALUE USED

DSN1940I DSN1COMP COMPRESSION REPORT
  HARDWARE SUPPORT FOR HUFFMAN COMPRESSION IS AVAILABLE
```

	UNCOMPRESSED	Estimated state Compressed FIXED	Estimated state Compressed HUFFMAN	Calculated Compressed from INPUT DICTIONARY
DATA (IN KB)	329,777	136,431	119,264	143,008
PERCENT SAVINGS		58%	63%	56%
AVERAGE BYTES PER ROW	163	69	61	72
PERCENT SAVINGS		57%	62%	56%
DATA PAGES NEEDED	91,305	38,182	33,334	39,623
PERCENT DATA PAGES SAVED		58%	63%	56%
DICTIONARY PAGES REQUIRED	0	32	32	32
ROWS SCANNED TO BUILD DICTIONARY		921	921	N/A
ROWS SCANNED TO PROVIDE ESTIMATE		2,100,000	2,100,000	N/A
DICTIONARY ENTRIES		4,096	4,080	4,096
TOTAL PAGES (DICTIONARY + DATA)	91,305	38,214	33,366	39,655
PERCENT SAVINGS		58%	63%	56%

```
DSN1994I DSN1COMP COMPLETED SUCCESSFULLY,        37,452  PAGES PROCESSED
```

Input file compression information
Depends on the input file, message may vary such as

```
INPUT DATA SET CONTAINS COMPRESSED DATA
USING HUFFMAN COMPRESSION TYPE
INPUT DICTIONARY BUILT BY INSERT
```

```
INPUT DATA SET CONTAINS non-COMPRESSED
DATA
```

IBM Confidential

Example of DSN1COMP – 1

```
DSN1998I INPUT DSNAME = DB2SMS.DSNDBC.DSN03545.ZZINSRGR.I0001.A001   , VSAM
DSN1944I DSN1COMP INPUT PARAMETERS

         INPUT DATA SET CONTAINS COMPRESSED DATA
          USING FIXED-LENGTH COMPRESSION TYPE
          INPUT DICTIONARY BUILT BY INSERT

           4,096  DICTIONARY SIZE USED
               0  FREEPAGE VALUE USED
               5  PCTFREE VALUE USED
            COMPTYPE(HUFFMAN) REQUESTED
              NO ROWLIMIT WAS REQUESTED
              ESTIMATE BASED ON DB2 LOAD METHOD
             255  MAXROWS VALUE USED
DSN1940I DSN1COMP COMPRESSION REPORT
  HARDWARE SUPPORT FOR HUFFMAN COMPRESSION IS AVAILABLE
```

Different COMPTYPE request
Results in different table layout

	UNCOMPRESSED	Estimated state Compressed HUFFMAN	Calculated Compressed from INPUT DICTIONARY
DATA (IN KB)	329,777	119,264	143,008
PERCENT SAVINGS		63%	56%
AVERAGE BYTES PER ROW	163	61	72
PERCENT SAVINGS		62%	56%
DATA PAGES NEEDED	91,305	33,334	39,623
PERCENT DATA PAGES SAVED		63%	56%
DICTIONARY PAGES REQUIRED	0	32	32
ROWS SCANNED TO BUILD DICTIONARY		921	N/A
ROWS SCANNED TO PROVIDE ESTIMATE		2,100,000	N/A
DICTIONARY ENTRIES		4,080	4,096
TOTAL PAGES (DICTIONARY + DATA)	91,305	33,366	39,655
PERCENT SAVINGS		63%	56%

```
DSN1994I DSN1COMP COMPLETED SUCCESSFULLY,          37,452  PAGES PROCESSED
```

IBM Confidential

Example of the DSN1COMP output – Input is non-compressed data

```
DSN1999I START OF DSN1COMP FOR JOB T5E11004 STEP2
DSN1998I INPUT DSNAME = DSNC000.DSNDBC.DB1.TS1.I0001.A001          , VSAM
DSN1944I DSN1COMP INPUT PARAMETERS

             INPUT DATA SET CONTAINS NON-COMPRESSED DATA

         4,096  DICTIONARY SIZE USED
             0  FREEPAGE VALUE USED
             5  PCTFREE VALUE USED
                COMPTYPE(ALL) REQUESTED
                NO ROWLIMIT WAS REQUESTED
                ESTIMATE BASED ON DB2 LOAD METHOD
           255  MAXROWS VALUE USED

DSN1940I DSN1COMP COMPRESSION REPORT
  HARDWARE SUPPORT FOR HUFFMAN COMPRESSION IS AVAILABLE
```

	UNCOMPRESSED	Estimated state Compressed FIXED	Estimated state Compressed HUFFMAN
DATA (IN KB)	1,758	1,232	1,278
PERCENT SAVINGS		29%	27%
AVERAGE BYTES PER ROW	38	28	29
PERCENT SAVINGS		26%	23%
DATA PAGES NEEDED	496	365	379
PERCENT DATA PAGES SAVED		26%	23%
DICTIONARY PAGES REQUIRED	0	16	20
ROWS SCANNED TO BUILD DICTIONARY		5,311	5,311
ROWS SCANNED TO PROVIDE ESTIMATE		50,000	50,000
DICTIONARY ENTRIES		4,096	4,080
TOTAL PAGES (DICTIONARY + DATA)	496	381	399
PERCENT SAVINGS		23%	19%

```
DSN1994I DSN1COMP COMPLETED SUCCESSFULLY,          475  PAGES PROCESSED
```

IBM Confidential

Summary

- Huffman compression introduced in Db2 12 Function Level (FL) 504
- Just like Lempel-Ziv, Huffman is a dictionary-driven compression and decompression algorithm
  - Dictionary pages are saved in the page set
- Hardware Prerequisites
  - z14 for the improved CMPSC instruction
- Fixed-Length vs Huffman compression can be selected by
  - Db2 12 Function Level 504 – subsystem parameter (ZPARM) TS_COMPRESSION_TYPE
  - Db2 12 Function Level 509 – more granular control at the table space/partition level
    - ✓Catalog Level 509 and APPLCOMPAT 509
- Db2 Catalog updated with more compression information
  - SYSIBM.SYSTABLEPART.COMPRESS and SYSTABLESPACE.COMPRESS is updated
  - SYSIBM.SYSTABLEPART.COMPRESS_USED is added

Summary

- DSN1COMP enhancement
  - Provides compression estimation for different compression algorithm (PH19242)
  - New COMPTYPE parameter (PH19242)
  - Input file can contain either compressed or non-compressed data (PH34808)
- Wide range of variability in terms of CPU and elapsed time performance
  - Different workload and different application usage affects the performance result
- Recommendation
  - The use of Huffman compression should be evaluated object by object
  - Implement in a controlled and incremental rollout using Db2's new object-level control

Db2 Huffman Compression Usability

David Nguyen – IBM nguyend@us.ibm.com

Frances Villafuerte – IBM francesv@us.ibm.com