IBM Performance Report

Performance Evaluation of Huffman Compression with SAP Banking on IBM Z

Authors:

Brenda Beane
Seewah Chan
Dr. Paul Lekkas
Veng Ly

Document Owner:

Veng Ly
SAP on IBM Z Performance

Version: 1.00
Date: July 28, 2020
Filename: Huffman_with_SAP_on_IBM_Z.pdf

Table of Contents

Figures

Tables

Disclaimers

Neither this document nor any part of it may be copied or reproduced in any form or by any means or translated into another language, without the prior consent of the IBM Corporation. IBM makes no warranties or representations with respect to the content hereof and specifically disclaims any implied warranties of merchantability or fitness of any particular purpose. IBM assumes no responsibility for any errors that may appear in this document. The information contained in this document is subject to change without any notice. IBM reserves the right to make any such changes without obligation to notify any person of such revision or changes. IBM makes no commitment to keep the information contained herein up to date.

Performance data and customer experiences for IBM and non-IBM products and services contained in this document were derived under specific operating and environmental conditions. The actual results obtained by any party implementing any such products or services will depend on a large number of factors specific to such party's operating environment and may vary significantly. IBM makes no representation that these results can be expected or obtained in any implementation of any such products or services.

The information in this document is provided "as-is" without any warranty, either expressed or implied. IBM expressly disclaims any warranties of merchantability, fitness for a particular purpose or infringement.

Trademarks

IBM, IBM logo, AIX, Db2, DS8800, FICON, FICON Express, OSA, OSA Express, HiperSockets, System Storage, IBM Z, zEC12, z13, z14, z15, z/Architecture, z/OS, z/VM, and zSeries, are trademarks or registered trademarks of the International Business Machines Corporation in the United States and other countries.

SAP, the SAP logo, and all other SAP product and service names mentioned herein are trademarks or registered trademarks of SAP SE in Germany and in several other countries all over the world.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

SUSE Linux Enterprise Server and SLES are registered trademarks of Novell, Inc., in the United States and other countries.

Other products may be trademarks or registered trademarks of their respective companies. Information is provided "AS IS" without warranty of any kind. Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, 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.

Other company, product, or service names may be trademarks or service marks of others.

Acknowledgements

The authors of this document would like to acknowledge the following individuals for their collaboration and support:

- Peter Becker, SAP on IBM Z development, IBM
- Nick Crimmins, z/OS system programmer, IBM
- Andrea Fuga, z/OS system programmer, IBM
- Akiko Hoshikawa, Db2 for z/OS development, IBM
- Regina Illner, SAP on IBM Z development, IBM

Feedback

Please send comments or suggestions for changes to vengly@us.ibm.com.

1.0 Introduction

Consumers and businesses are churning out a lot of data every moment of the day. Quintillions of bytes of data are created daily. This huge amount of data can become a valuable asset for business intelligence which can help business leaders with insights to steer decisions and to profit on the insights. With this massive data growth explosion, disk storage management in the data center can become a complex and expensive problem. Data compression can help in addressing this problem. Enhancing compression efficiency can further improve the situation even if data compression has been already deployed.

The IBM z14 processor introduced a new technique for compressing data [1]. It provides optimized on-chip compression and uses Huffman encoding to achieve better data compression efficiency than the prior fixed length algorithm. Huffman encoding is a variable-length encoding system that assigns smaller codes for more frequently used characters and larger codes for less frequently used characters in order to reduce the size of the data being compressed.

Starting with Db2 12 for z/OS on z14 or newer IBM Z processors, Huffman compression is supported for Db2 universal table spaces. Db2 for z/OS continues to support the fixed length compression algorithm. The SAP on Z solution has been using fixed length compression for its table spaces for many years. SAP on Z customers have been enjoying the benefits of compression for a long time.

The IBM SAP on Z Performance Team, located in Poughkeepsie, NY, conducted a study to see if there could be any space saving benefit or significant system performance impact for an SAP on Z database from the improved compression algorithm.

This study used SAP Banking Services (SBS), officially named “banking services from SAP”. The banking database had 100 million accounts. This is comparable to the number of accounts held by some of the largest banks in the world. This database was compressed using Huffman encoding and the space usage was compared to the same database compressed with fixed length encoding. The SAP Banking Account Settlement and Day Posting workloads were run to quantify the database growth and any impact to the system performance with Huffman compression compared to fixed length compression. Using both a batch and an OLTP workload gave a more complete picture of potential performance impacts since these workloads have different characteristics. Both these workloads are good representations of customer workloads. The measurements were conducted on an IBM z15 with Db2 12 for z/OS function level 504.

This paper documents these tests and findings. The measurements that were done were stress tests, not certified benchmarks.

2.0 Overview of the IBM Z Compression Enhancement

For many years now, SAP on Z has been using fixed length compression for table spaces, based on the Lempel-Ziv (LZ) tree encoding compression technique. The SAP on Z customers have been getting the benefits of compression for a long time. However, they may see additional benefits from the improved compression efficiency of the Huffman encoding.

The Huffman encoding uses the same LZ tree-based mechanism. However, it adds a symbol translation table which maps the 12bit fixed length LZ tree node index into a variable length (up to 16 bit) symbol. The Huffman encoding exploits the fact that certain data or characters are more frequent than others. For strings of data that frequently occur, a shorter symbol code is assigned; for strings that are infrequent, longer symbol codes are assigned. Thus, instead of always taking the 12 bits that are used in fixed length encoding, it takes fewer bits to compress the frequent data and more bits for infrequent ones.

To take advantage of the compression improvement, the minimum requirements are Db2 12 for z/OS function level 504 running on a z14 class machine and all newer IBM Z processors [2]. Db2 12 for z/OS added the new system parameter TS_COMPRESSION_TYPE. The value can be FIXED_LENGTH or HUFFMAN. FIXED_LENGTH is the existing compression support and default. In addition, Db2 utility DSN1COMP (APAR PH19242) was enhanced to provide compression ratio estimation for Huffman compression and provide comparison to existing fixed length compression.

The Huffman compression enhancement is very attractive to customers who use large tables with Db2 for z/OS. The SAP on Z customers running SAP core banking applications are good candidates to take advantage of this new compression enhancement. During system installation, SAP on Z has Db2 table spaces set to use fixed length (LZ) compression. By switching to Huffman compression, customers may see additional storage space saving benefits from the improved compression efficiency.

The compression ratio or storage space savings achieved depends on the characteristics of the data. A better compression ratio may reduce I/O costs and affect the system performance. The system performance impact also depends on the workload characteristics, whether the data access patterns are sequential as in batch processing or random as in OLTP.

To assess the space usage and system performance impact of employing Huffman compression as compared to the fixed length (LZ) compression in the SAP on Z environment, the following were evaluated:

- Size of the database space before and after running the SAP banking workloads
- System performance of a batch processing workload with SAP Banking Account Settlement
- System performance of an online transaction processing workload with SAP Banking Day Posting

The measurements were conducted on a z15 using the SAP core banking workloads with 100 million accounts, which is comparable to the number of accounts held by some of the largest banks in the world.

3.0 Test Workloads

Many of the SAP on Z customers are financial and banking institutions. It is fitting that the SAP core banking workloads are used for these tests. There are two distinct SAP banking workloads. One is the Account Settlement workload and the other is the interactive Day Posting workload.

3.1 SAP Account Settlement

The SAP Account Settlement workload performs account balancing on all accounts in the database. In this study, there were 100 million accounts in the database. Account balancing is done periodically for each account to calculate charges (fees) and interest as well as posting them. It is usually a series of batch jobs executed monthly or quarterly at night depending on the institution and the type of account. The jobs must complete within a critical period of time, typically called the batch processing window. The data for each account is processed once. The workload is insert/update intensive and its data access pattern is mostly sequential.

This workload is part of the SBS application suite. There is an average of 20 items per account posted in a calendar month, which is on average 20 business days. The key metric is the overall elapsed time of the batch jobs.

3.2 SAP Day Posting

The SAP Day Posting workload simulates the daytime business transactions, called postings, of a retail bank. A typical example of a posting is a payment out of the account or a deposit into the account. The SAP Day Posting workload is an online transaction processing (OLTP) workload. It invokes simple to moderate SQLs and has a random data access pattern.

This workload is part of the SBS application suite. The workload consists of interactive “users” going through repetitive cycles of 15 dialogue steps.

Operation steps:

1. Create a total of 150 postings via five BAPI calls
2. Create five postings
3. Create bank statement
4. Read postings for account
5. Read details of postings
6. Create five postings
7. Create one bank statement for account
8. Create five postings
9. Create one bank statement for account
10. Create five payment orders
11. Read balances of account
12. Create five postings
13. Create one bank statement for account
14. Read balances for account
15. Read master data for account

4.0 Test Environment

Generally, the SAP environment has three tiers: the presentation server, the application server, and the database server. The presentation server is a client machine running the GUI window. It is generally remote to the application server and the database server. This test environment was configured with both the SAP database server and the application servers co-located on a single IBM Z machine.

The following diagram, Figure 1, is a logical representation of the test environment with a z/OS LPAR running the SAP database server and four Linux LPARs running the SAP application servers. All servers communicated via HiperSockets which is an intra-CEC cross memory connection for low network latency.

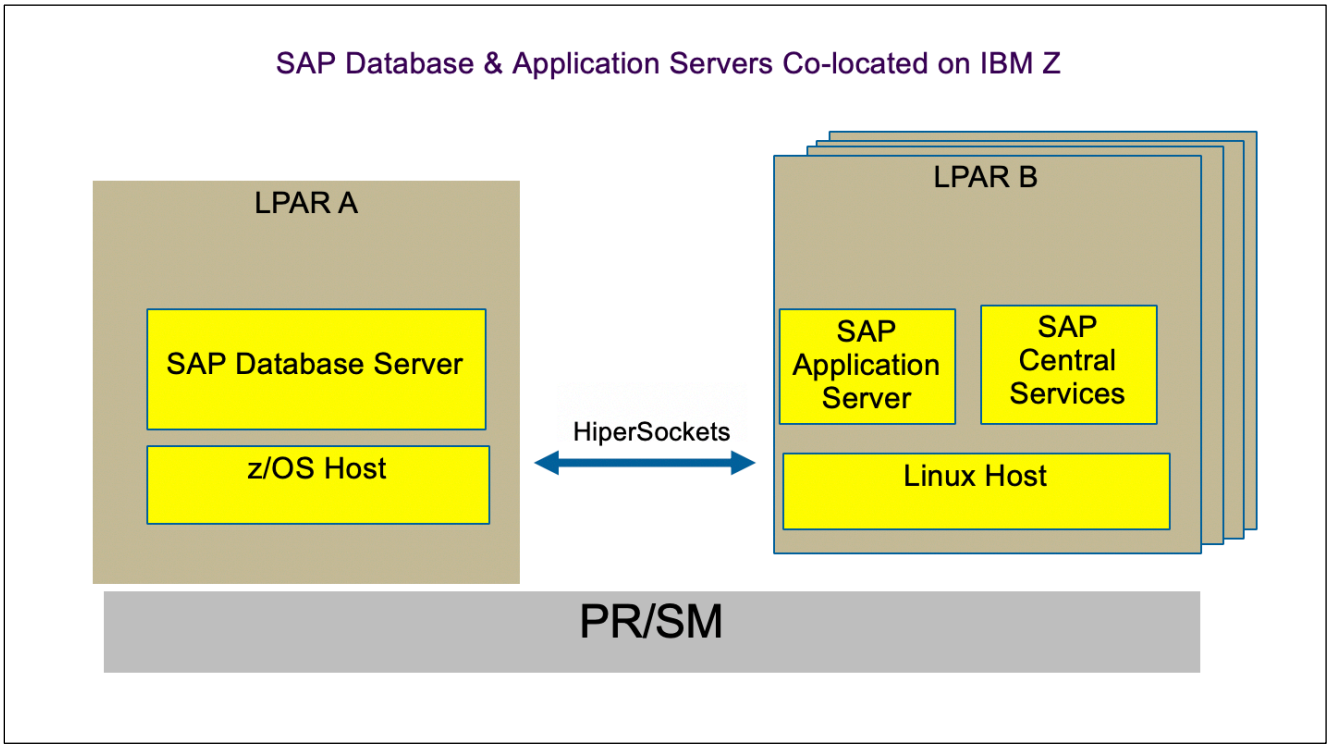

Figure 1: Logical Representation of Test Environment

4.1 Hardware Environment

An IBM z15 was used in this test environment. Attached to the z15 was an IBM storage DS8886 which housed the 100 million account banking database. The SAP application servers and the database server were co-located in the single z15. Figure 2 is the representation of the hardware test environment.

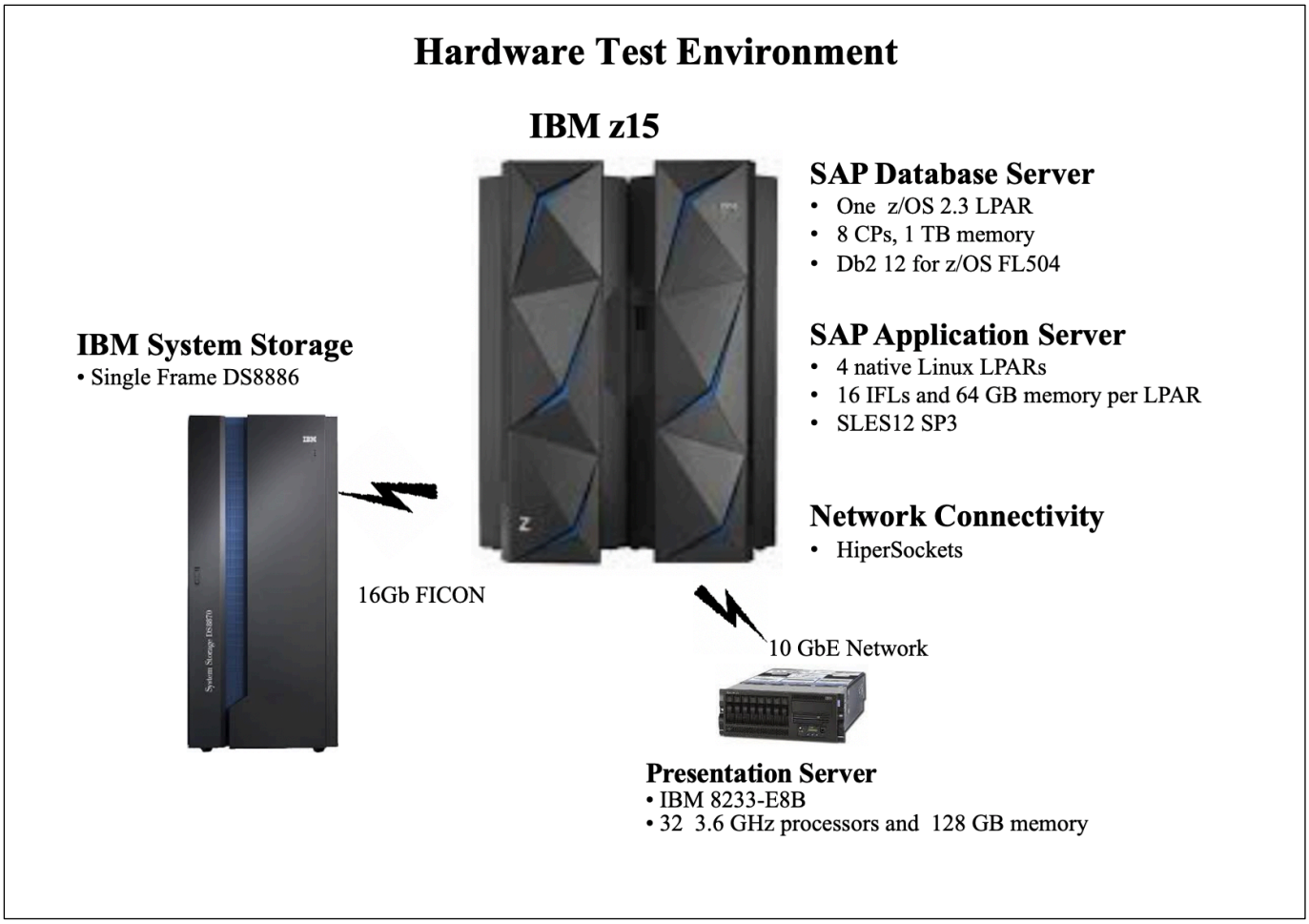

Figure 2: Hardware Test Environment

SAP Database Server
The SAP database server operated within an LPAR on an IBM z15. The LPAR was configured with 8 dedicated CPs and 1TB of memory. The minimal Z processor requirement for Huffman compression is z14.

Database Storage
An SBS 9.0 database with 100 million accounts was used for this test. It resided on an IBM System Storage DS8886 subsystem, spread across 310 3390-mod54 volumes. Two flash copies of the database were maintained. One was with fixed length compression and the other one was with Huffman compression.

SAP Application Servers
The SAP application servers were on four LPARs on an IBM z15. Each LPAR had 16 dedicated IFLs and 64GB of memory.

SAP Presentation Servers
The SAP presentation server was on a p5 class Power machine with 32 3.6 GHz processors and 128GB of memory.

Network
No physical network is required between the SAP application servers and the SAP database server since they are co-located on a single Z CEC. HiperSockets (intra CEC cross memory) communication was used. Network communication between the SAP presentation server to the SAP application servers was via a 10GbE connection to initiate the workload run.

4.2 Software Environment

z/OS
z/OS release 2.3

Db2 for z/OS
Db2 12 FL504 (the minimum requirement for Huffman compression)

Db2 Connect
IBM Data Server Driver for CLI that is shipped as part of Db2 Connect 11.01

LINUX
SLES12 SP3

SAP
SAP NetWeaver 7.5 SP6
Banking Services from SAP 9.0 SP4
SAP Kernel 7.49 level 401

4.3 Database

The SBS 9.0 banking database used in this test contains 100 million accounts. It is comparable to the number of accounts held by some of the largest banks in the world. The active database size is more than 16 TB.

In general, SAP databases have a very large number of database objects, which include table spaces and indexes. This translates into a very large number of datasets to back the database. According to the SAP Database Administration Guide [3], a large number of tables remain empty for most SAP systems. To reduce the number of datasets needed for the database, table spaces and indexes are created with the Db2 attribute DEFINE=NO during the SAP system installation and upgrade processes. This prevents the underlying datasets from being created until the first row is inserted into a table. However, there are still tens of thousands of datasets for an SAP database.

The SAP database used in this test used universal table spaces (UTS). Huffman compression is only supported for universal table spaces. The database was built using the implicit object creation of Db2, which means that every table was contained in a partition-by-growth universal table space (PBG UTS) after the SAP system installation. The key banking tables were converted to partition-by-range universal table spaces (PBR UTS), as recommended by SAP.

For many years now, the SAP on Z solution has been using fixed length compression for table spaces. So, our SAP customers have been getting the benefits of compression for a long time. They may see additional benefits from an improved compression algorithm.

There are more indexes than table spaces in a typical SAP on Z database. However, the space used within a typical SAP database is generally split 50/50 between table spaces and indexes. The SAP banking database follows this same pattern. The key banking tables account for close to 95% of the total table space size in the database.

5.0 Test Methodology

The objective for this study was to evaluate Huffman compression in terms of space usage and system performance by comparing it with fixed length compression. Both the SAP Account Settlement and Day Posting workloads were run with fixed length compression and Huffman compression. The fixed length compression measurements were the baseline measurements.

5.1 Preparing the Database

For this study, two copies of the SBS 9.0 100 million account database were prepared. They were identical except one had table spaces compressed with fixed length compression and the other had table spaces compressed with Huffman compression.

Since the SAP Account Settlement and Day Posting workloads were used to evaluate the impact of compression on system performance, all the table spaces used by these workloads were converted to the Huffman compression algorithm. The IBM SAP on Z Performance Team has been running these workloads for many years and is very familiar with the table spaces used by each. Nineteen of these table spaces are partition-by-range universal table spaces with 512 partitions. In total, over 10,000 datasets were converted to Huffman compression. The table spaces converted to Huffman compression accounted for close to 95% of the total table space size in the database.

Initially, the database had all its table spaces compressed with fixed length compression. By default, this is a result of the SAP system installation. Therefore, all the table spaces already had the COMPRES YES attribute set. To ensure a fair comparison between fixed length and Huffman compression, the REORG utility was run against the fixed length compressed table spaces used by these workloads to reclaim fragmented space and re-establish clustering order. Since REORGs were being done, the fixed length compression dictionary was rebuilt by not specifying the KEEPDICTIONARY parameter. The Db2 system-parameter TS_COMPRESSION_TYPE was set to FIXED_LENGTH when these REORGs were done.

The Huffman compressed database was created from the fixed length compressed database by converting the table spaces used by the banking workloads to Huffman compression. The Db2 system parameter TS_COMPRESSION_TYPE was set to HUFFMAN and the REORG utility was run without specifying the KEEPDICTIONARY parameter on each table space to be converted to Huffman compression. In this case, the REORG utility builds a Huffman compression dictionary, along with reclaiming fragmented space and re-establishing the clustering order.

To verify the type of compression in a particular table space, the DSN1PRNT utility was used. The LZDHTYPE field is set to either 'F' for fixed length compression or 'H' for Huffman compression. The following is a snippet of a DSN1PRNT output for the Huffman compression.

```
PAGE: # 00000002 ------------------------------------------------------------------------------
LZDH INFO  LZDHIDFR='2370'X  LZDHLENG=1024  LZDHVID='LZDH'  LZDHTYPE='H'  LZDHPART=1
           LZDHFLGS='F00000'X  LZDHSTOR=77728  LZDHESTG=0  LZDHNOFF=0  LZDHNODE=4003  LZDHSIZE=4080
           LZDHNUMP=19  LZDHNUDP=10  LZDHEGRO='000A0100'X  LZDHCGRO='000B0000'X
           LZDHVERS='0000000000000000'X  LZDHDBNM='TRBK4014'  LZDHTSNM='BCARPAYM'
```

5.2 Key Performance Indicators (KPIs)

To gauge the system resource requirements and to provide performance insights and comparisons, the following Key Performance Indicators (KPIs) were captured for every measurement in the test scenarios:

- CPU utilization on the database server and the application server
- Elapsed time of the batch processing in the Account Settlement workload
- Average response time of transactions in the Day Posting workload
- External Throughput Rate (ETR)
- Internal Throughput Rate (ITR)
- Db2 NPAGES from RUNSTATS

For the Account Settlement workload, the elapsed time is the most important KPI since many customers must complete this work within a batch processing window.

For the Day Posting workload, the average response time is one of the important KPIs to ensure end user satisfaction and productivity.

The External Throughput Rate (ETR) is the transaction rate. For the Account Settlement workload, this is 100 million accounts divided by the maximum batch elapsed time. For the Day Posting workload, it is the number of postings per measurement time interval.

The Internal Throughput Rate (ITR) is the ETR normalized or projected to 100% processor utilization. ITR provides a reliable basis for measuring processor capacity [4].

Db2 NPAGES is the KPI used in this study to quantify the database space used. It is discussed in detail in the next section.

5.3 Collecting Storage Space Statistics

After running the initial REORGs to create the compression dictionaries for each compression type, RUNSTATS was run on the table spaces to gather summary information about the characteristics of the data in the table spaces.

To quantify the data space usage between fixed length and Huffman compression, the metric of NPAGES from the Db2 SYSTABLES catalog table was used. This field gets updated by RUNSTATS. According to the Db2 SQL Reference, NPAGES is defined as the total number of pages that include rows of the table. Since the space used by the user data is of interest, it makes sense to focus on the pages that actually contain user data. It is important to measure the space usage by pages since Db2 stores data on a page basis. Table spaces and index spaces contain pages. Pages are moved in to/out of buffer pools. To achieve space savings with compression, the number of pages must be reduced, not just the length of the individual rows.

The goal for using any type of data compression is to reduce the overall disk space required for data. So, many people initially think to look at the amount of space being used for the datasets on disk as a metric. This isn't a good approach since the size of a dataset depends on its primary and secondary space allocations, in Db2 terms, its PRIQTY and SECQTY attributes. If these allocations are manually specified and significantly large, then it is possible to reduce the number of Db2 data pages via compression without seeing a corresponding decrease in allocated space.

Space statistics were also collected again after running the workloads. This allowed the calculation of the compression statistics for the tables that start empty, as well as the growth rate of the key banking tables. To be consistent, after running each workload, REORGs and then RUNSTATS were run before extracting the space statistics.

6.0 Measurement Results and Analysis

Measurements were obtained to assess the space usage and the system performance impact of Huffman compression compared to fixed length compression using the SAP Account Settlement and Day Posting workloads. Storage space statistics were collected before and after running the workloads.

6.1 Space Analysis

6.1.1 SAP Account Settlement

The Account Settlement workload processed and populated large amounts of data, resulting in a significant database space growth. As time progresses, the space growth can become a problem. Thus, storage space reduction with better compression efficiency from the Huffman encoding becomes very important and attractive for large SAP banking databases. Table 1 shows the storage space statistics in NPAGES for the key database tables used by the SAP Account Settlement workload. All tables used by the SAP Account Settlement were compressed, both banking and non-banking tables. However, it is too numerous to list all of them. Thus, only the key tables with significantly large sizes are listed. These tables represent a pretty good cross-section of the data and had the most impact on the overall database space savings. A few of them started empty in these tests, and they were populated by the banking workload.

Compression Type	**Fixed Length**		**Huffman**		
Mesurement runID	**S00315G1**		**S00314G1**		
Database State	**before**	**after**	**before**	**after**	**delta**
Table Name					
BCA_ACCTBAL	1,179,611	1,179,611	965,838	965,838	-18%
BCA_BCAS_DUE	1,334,331	1,334,331	1,093,450	1,093,450	-18%
BCA_BCAS_EVBST	1,808,523	1,808,523	1,499,212	1,499,212	-17%
BCA_CN_EVENT	7,314,059	7,316,962	6,133,836	6,133,979	-16%
BCA_CN_LINK	9,313,988	9,313,988	9,313,988	9,313,988	0%
BCA_CN_PER_ACBAL	806,044	806,044	699,111	699,111	-13%
BCA_CNSP_ACCT	1,817,773	1,817,773	1,614,307	1,614,307	-11%
BCA_COUNTER	26,597,090	26,597,090	19,252,151	19,252,151	-28%
BCA_CONTRACT	961,489	961,489	833,114	833,114	-13%
BCA_GL_PAYMITEM	44,368,244	62,318,105	30,098,775	45,644,883	-27%
BCA_PAYMITEM	87,549,480	115,872,734	38,593,611	67,794,851	-41%
BCA_TRANSFIG	15,581,200	15,581,200	13,509,935	13,509,935	-13%
ADCP	1,297,310	1,297,310	1,236,547	1,236,547	-5%
ADRC	1,577,785	1,577,785	908,284	908,284	-42%
ADRP	2,572,192	2,572,192	1,738,753	1,738,753	-32%
BANK_RECMN_RECR	1,912,026	1,912,026	1,574,643	1,574,643	-18%
BUT000	2,691,567	2,691,567	1,737,013	1,737,013	-35%
BUT020	924,022	924,022	819,233	819,233	-11%
BUT021_FS	621,106	621,106	554,306	554,306	-11%
BCA92	0	1,947,573	0	1,451,660	-25%
BKK92_POSTINGS	0	1,386,083	0	1,178,505	-15%
BKK92_SUMS	0	5,191,483	0	2,178,544	-58%
BCA96	0	61,411,486	0	27,042,824	-56%
BCA92_RESTART	0	3,475,621	0	3,357,490	-3%
total	210,227,840	329,916,104	132,176,107	212,132,621	-36%
total w/o BCA_PAYMITEM and BCA96	122,678,360	149,156,263	93,582,496	113,937,456	-24%

Table 1: SAP Account Settlement Database Table Space Statistics (NPAGES)

Huffman compression provided a significant space savings compared to fixed length compression, specifically it used 36% fewer NPAGES. The rightmost column in the above table shows the NPAGES delta after the workload run between the two compression encodings. BCA_PAYMITEM and BCA96 are two of the largest tables with the highest growth in terms of NPAGES. They are banking tables that compressed better than a typical Db2 table. The compression ratios for BCA_PAYMITEM and BCA96 are 41% and 56% respectively. If these two tables are excluded from the total NPAGES usage then the space savings with Huffman compression is 24%, which is more in line with typical Db2 space savings seen with Huffman compression. BCA_PAYMITEM and BCA96 are often cited by our banking customers as the largest key tables. Given the excellent compression ratios seen in this study, they may be very interested in exploring Huffman compression.

The Huffman compression ratio is highly dependent on the characteristics of the data. There are some scenarios where Huffman compression may not provide a reduction in NPAGES. In general, if rows are short and the page contains close to the Db2 limit of rows per page, then compressing the rows shorter will not allow more rows to be stored on the page. Another case is the opposite example when there are very long or large rows. The compression savings may not be sufficient enough to fit more large rows on the page. In this situation, to get the compression space saving benefit, the table space may need to be converted to a larger page size to fit more rows per page. For example, a 4K table could be converted to an 8K, 16K, or 32K page size. There are other table space attributes that greatly influence the space usage. For example, MAXROWS or high PCTFREE specifications can limit the number of rows per page regardless of the shorter row length due to compression.

While processing 100 million accounts, the SAP Account Settlement workload inserts a significant number of records into the database. Table 1 shows BCA_PAYMITEM and BCA96 as the heaviest inserted tables. BCA_GL_PAYMITEM, BCA92, BKK92_POSTINGS, BKK92_SUMS, and BCA_RESTART also get inserted into.

Figure 3 shows the space used for these tables grew by 120 million NPAGES for fixed length compression and 80 million NPAGES for Huffman compression. The growth rate is lower for Huffman compression which may ease some of customer's long-term database growth concerns.

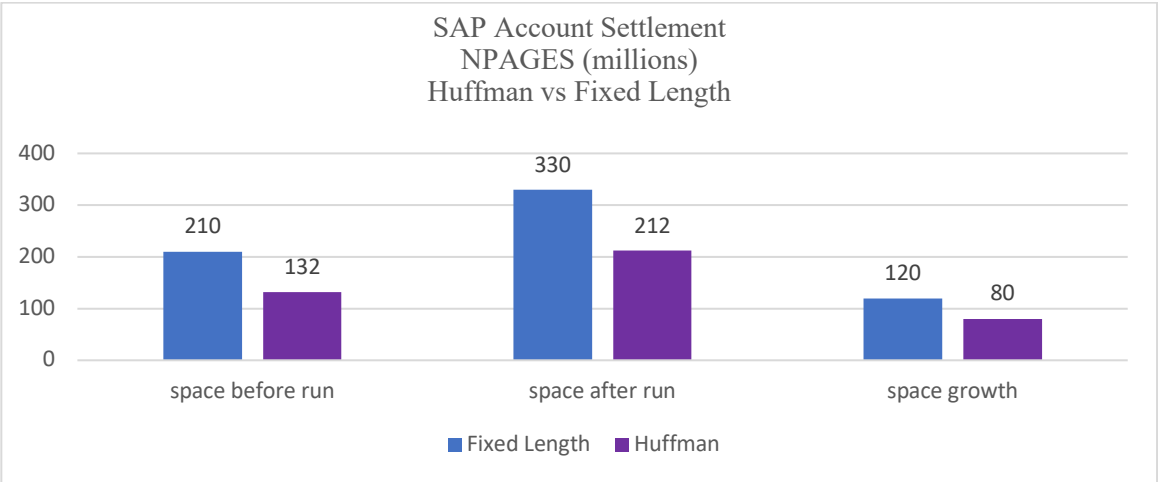

Figure 3: SAP Account Settlement Space Statistics

The actual overall database space savings with Huffman compression in this test was 18%. This is because, in a typical SAP application environment as in our SAP banking environment, the space used within our banking database is approximately split 50/50 between table spaces and indexes. Thus, the space saving for the entire database is half of the 36% measured for table space savings, which is 18%. In a non-SAP application environment where there are fewer indexes and much less space used for indexes, the overall database space savings could be more.

6.1.2 SAP Day Posting

SAP banking databases are typically very large in customer shops due to space growth from daily postings. This certainly makes Huffman encoding with better compression efficiency very attractive. Table 2 shows the storage space statistics in NPAGES for the key database tables used by the SAP Day Posting workload. All tables used by the SAP Day Posting workload were compressed, both banking and non-banking tables. It is too numerous to list all of them. Thus, only the key tables with significantly large sizes are listed. They represent a good cross-section of the data and had the most impact on the overall database space savings. A few of them started empty in these tests, and they were populated by the banking workload.

Compression Type	Fixed Length		Huffman		
Mesurement runID	S00130G1		S00201G1		
Database State	before	after	before	after	delta
Table Name					
BCA_ACCTBAL	1,179,611	1,179,611	965,838	965,838	-18%
BCA_BCAS_DUE	1,334,331	1,352,106	1,093,450	1,104,691	-18%
BCA_BCAS_EVBST	1,808,523	1,917,019	1,499,212	1,589,823	-17%
BCA_CN_EVENT	7,314,059	7,316,971	6,133,836	6,133,971	-16%
BCA_CN_LINK	9,313,988	9,313,988	9,313,988	9,313,988	0%
BCA_CN_PER_ACBAL	806,044	806,044	699,111	699,111	-13%
BCA_CNSP_ACCT	1,817,773	1,817,773	1,614,307	1,614,307	-11%
BCA_COUNTER	26,597,090	26,597,090	19,252,151	19,252,151	-28%
BCA_CONTRACT	961,489	961,489	833,114	833,114	-13%
BCA_GL_PAYMITEM	44,368,244	45,749,851	30,098,775	31,151,976	-32%
BCA_PAYMITEM	87,549,480	90,795,428	38,593,611	41,563,594	-54%
BCA_TRANSFIG	15,581,200	15,581,200	13,509,935	13,509,935	-13%
ADCP	1,297,310	1,297,310	1,236,547	1,236,547	-5%
ADRC	1,577,785	1,577,785	908,284	908,284	-42%
ADRP	2,572,192	2,572,192	1,738,753	1,738,753	-32%
BANK_RECMN_RECR	1,912,026	1,912,026	1,574,643	1,574,643	-18%
BUT000	2,691,567	2,691,567	1,737,013	1,737,013	-35%
BUT020	924,022	924,022	819,233	819,233	-11%
BUT021_FS	621,106	621,106	554,306	554,306	-11%
BCA_PO_HD	0	40,589	0	24,483	-40%
BCA_PO_IT	0	87,161	0	46,207	-47%
BCA_PO_NT	0	16,361	0	11,288	-31%
BCA_PO_ERR_IDX	0	28,006	0	18,192	-35%
total	210,227,840	215,156,695	132,176,107	136,401,448	-37%
total w/o BCA_PAYMITEM	122,678,360	124,361,267	93,582,496	94,837,854	-24%

Table 2: SAP Day Posting Database Table Space Statistics (NPAGES)

Huffman compression provided a significant space savings compared to fixed length compression, specifically it used 37% fewer NPAGES. The rightmost column in the above table shows the NPAGES

delta after the workload run between the two compression encodings. BCA_PAYMITEM is the largest table in terms of NPAGES. It is a banking table that compressed better than a typical Db2 table. The compression ratio for BCA_PAYMITEM is 54%. If this table is excluded from the total NPAGES usage, then the space savings with Huffman compression is 24%, which is more in line with typical Db2 space savings seen with Huffman compression. BCA_PAYMITEM is often cited by our banking customers as one of the largest key table. Given its excellent compression ratio seen in this study, they may be very interested in exploring Huffman compression.

The storage space growth rate for the SAP Day Posting workload was 20% less with Huffman compression compared to fixed length compression. Figure 4 shows that after completing the SAP Day Posting workload run, the storage space growth was 4 million NPAGES with Huffman compression and 5 million for fixed length compression. It is important to point out that the test measurement interval peak load period was only about 30 minutes. A typical customer would run peak load for several hours daily and have more data growth.

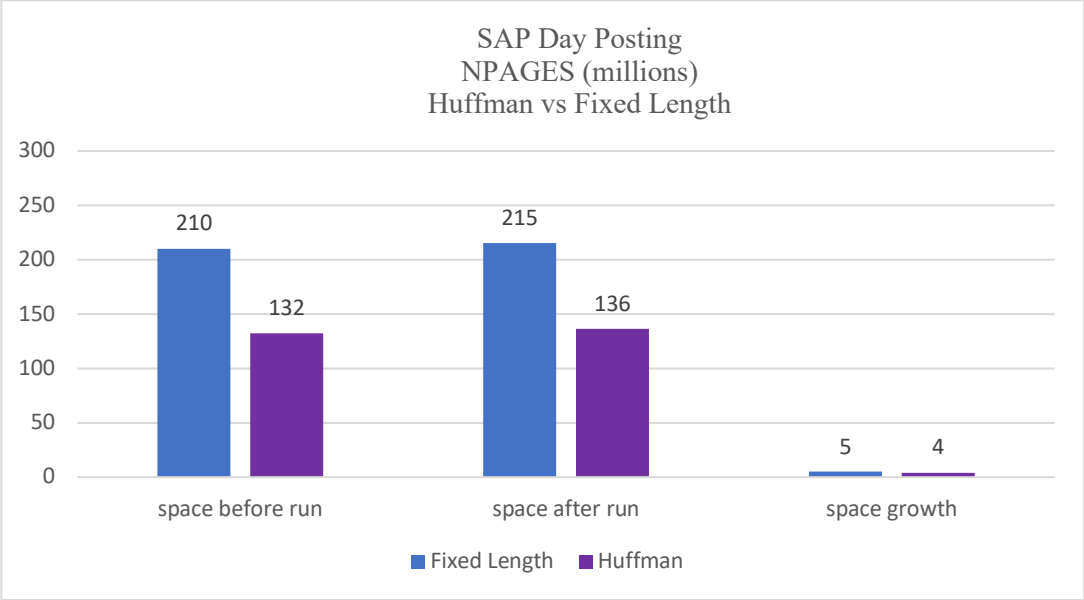

Figure 4: SAP Day Posting Space Statistics

The actual overall database space savings in this test was 18%. This is because, in a typical SAP application environment as in our SAP banking environment, the space used within our banking database is approximately split 50/50 between table spaces and indexes. Thus, the space saving for the entire database is half of the 37% measured for table space savings, which is about 18%. In a non-SAP application environment where there are fewer indexes and much less space used for indexes, the overall database space savings could be more.

The 18% space savings are the same for both the SAP Day Posting and Account Settlement workloads. This makes sense since the compression ratio is based on the characteristics of the data and many of the same tables are used by both workloads. In short, the SAP banking environment may be a suitable candidate to take advantage of this enhanced compression.

6.2 System Performance

6.2.1 SAP Account Settlement

After finding a space saving benefit of 18% with Huffman compression, the big question is "Is there a performance benefit?" For the Account Settlement workload, the answer is yes. The following is detailed discussion and analysis of the system performance measurements with this workload.

The Account Settlement workload is a batch processing workload, processing 100 million accounts in this study. The number of parallel batch jobs was configured and held constant at 128, which is in the reasonable range for a large batch load. The batch load was distributed equally among the four SAP application servers running on native Linux systems with four SAP instances per Linux system. The SAP application servers and the database server were co-located on a single z15 machine, communicated via HiperSockets connectivity. The Linux application server configuration had a total of 64 IFL processors. The z/OS LPAR configuration had 8 general purpose processors. The same configuration was used for the Huffman and fixed length compression tests.

For Account Settlement processing, the most important metric is the elapsed time, addressing the critical batch processing window. Shorter elapsed time is better. The following figure, Figure 5, shows the ratio of the batch elapsed time comparison of Huffman versus fixed length compression. Huffman compression had a 6% shorter batch elapsed time as compared to the fixed length compression baseline.

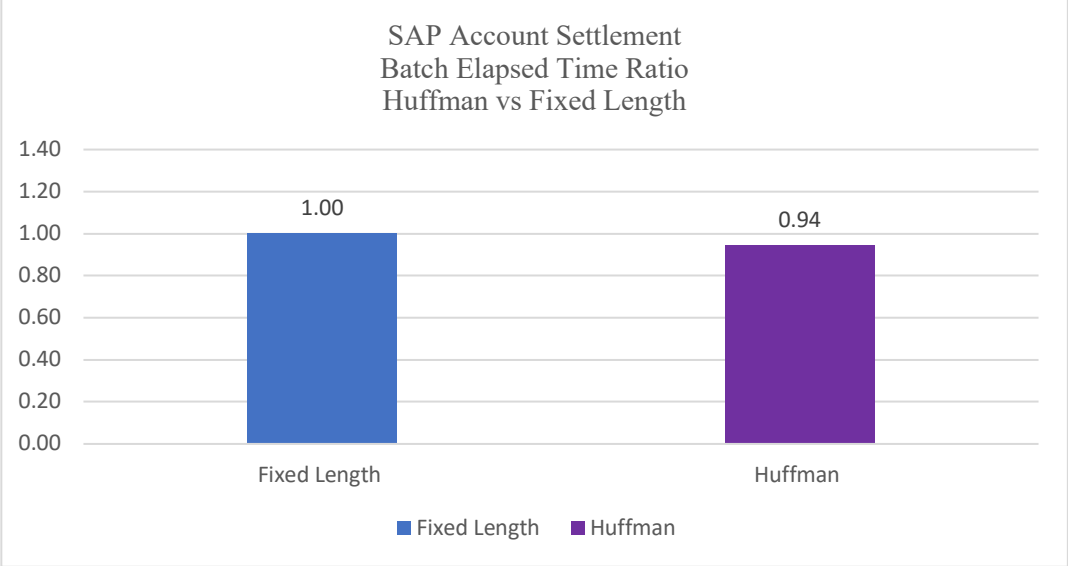

Figure 5: SAP Account Settlement Batch Elapsed Time Ratio

Another important key metric is the ITR of the SAP database server. There was a 3% improvement in ITR with Huffman compression versus the fixed length compression baseline. Figure 6 shows the ITR ratio for the SAP database server.

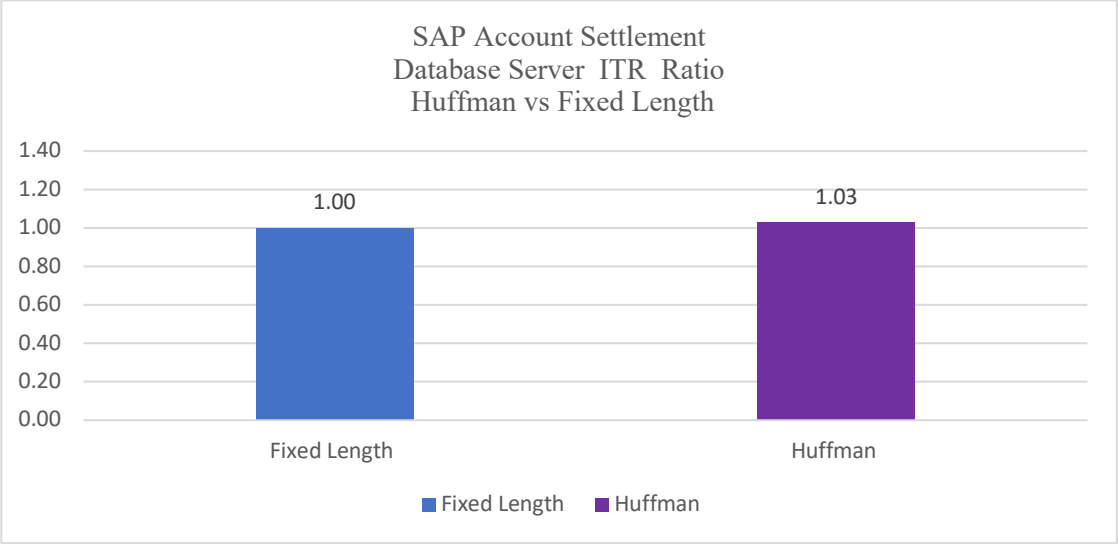

Figure 6: SAP Account Settlement Database Server ITR Ratio

The performance improvement seen with the SAP Account Settlement workload was due to its mostly sequential data access characteristics. It benefited from the better Huffman compression ratio by having more records stored in a data page. This reduced the number of getpages and data page I/Os, as indicated in Table 3 below. The extracted Db2 statistics reports show the number of prefetch and synchronous reads which are physical database I/Os. They were less for the Huffman encoding as compared to the fixed length baseline. This I/O reduction translated directly to savings in CPU cycles consumed by the account settlement workload with Huffman compression.

Compression	**Fixed Length**	**Huffman**	
Measurement runID	**S00315G1**	**S00314G1**	
READ OPERATION	**quantity**	**quantity**	**delta**
GETPAGE REQUEST	5384.8M	5321.4M	-1%
SYNCHRONOUS READS	53516.9K	52971.0K	-1%
SEQUENTIAL PREFETCH REQUEST	32968	27475	-17%
SEQUENTIAL PREFETCH READS	32925	27433	-17%
LIST PREFETCH REQUESTS	100.3M	100.3M	0%
LIST PREFETCH READS	10419.0K	3260.8K	-69%
DYNAMIC PREFETCH REQUESTED	105.4M	102.1M	-3%
DYNAMIC PREFETCH READS	19319.1K	18863.3K	-2%

Table 3: Db2 Statistics Report of Total Read Operations

To recap, the SAP Account Settlement workload had benefits from Huffman compression in both space savings and system performance improvement:

- 18% savings in storage space
- 6% improvement in elapsed time
- 3% improvement in database server ITR

6.2.2 SAP Day Posting

After finding a space saving benefit of 18% with Huffman compression, the big question is "Is there a performance benefit?" For the SAP Day Posting workload, the answer is no. The following is detailed discussion and analysis of the system performance measurements with this workload.

The SAP Day Posting workload is an interactive workload using dialog work processes, instead of batch processes to handle the application execution. The simulated user load was distributed evenly to four SAP application servers running on native Linux systems with four SAP instances per Linux system. The SAP application servers and the database server were co-located on a single z15 machine, communicated via HiperSockets connectivity. The Linux application server configuration had a total of 64 IFL processors. The z/OS LPAR configuration had 8 general purpose processors. The same configuration was used for the Huffman and fixed length compression tests.

There was a slight increase in CPU consumption with Huffman compression which is evident by a 4% degradation in ITR with Huffman compression versus the fixed length compression baseline. The SAP Day Posting workload has a more random data access pattern. This means that the required records are scattered in many pages in the table. Result rows are accessed directly by index. Db2 traverses through an index tree for the searched data record residing on the data page. Although the data records are compressed, the number of data pages accessed were about the same. The number of getpages and data page I/Os were not reduced. Thus, the measurements show that there was a small performance cost to trade off for significant space savings. Figure 8 shows the ITR ratio comparison for the database server.

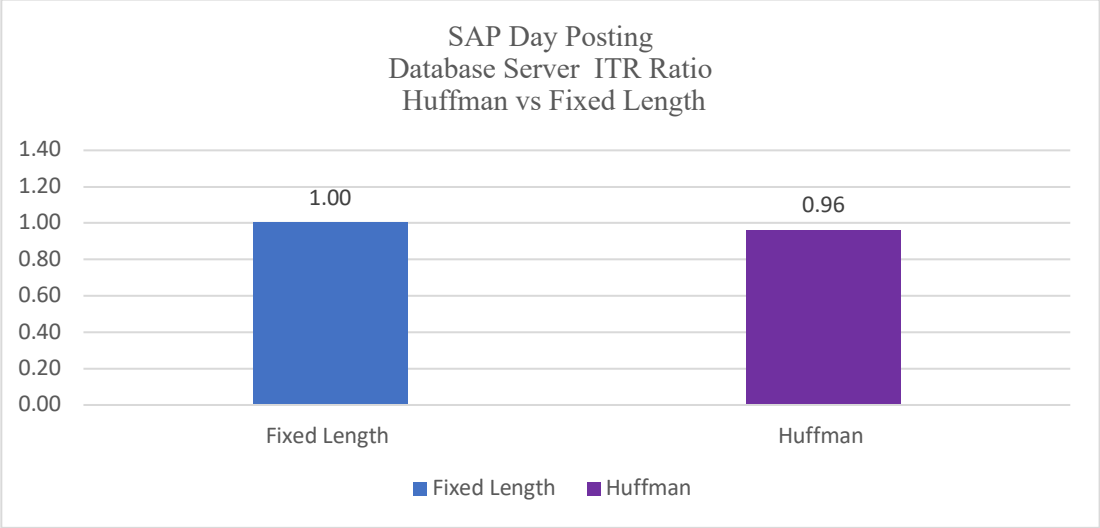

Figure 7: SAP Day Posting Database Server ITR Ratio

To recap, the SAP Day Posting workload had benefits from Huffman compression in space savings, but there was a slight degradation in system performance:

- 18% savings in storage space
- Equivalent average sub-second response times
- 4% degradation in database server ITR

7.0 Conclusion

To address the massive data growth explosion and its associated problems, including the cost of disk storage, IBM enhanced its compression technology to meet the ever-increasing data storage demand. It uses optimized on-chip compression with Huffman encoding in z14 and newer IBM Z processors to improve the compression efficiency to help manage disk storage space usage.

The Huffman encoding algorithm can provide a better compression ratio for data with a high frequency of occurrences because it takes fewer bits to compress the frequent data than the 12bit fixed length compression algorithm.

In this study, to assess the storage space usage and system performance of using Huffman compression, a series of large-scale measurements were conducted on z15 with the standard SAP core banking application workloads with a 100 million account database.

Our measurements show that for our SAP banking database Huffman compression reduced the space used for the table spaces by 37% compared to fixed length compression. However, the overall database space savings with Huffman compression is 18%. This is because Huffman compression only applies to table spaces and the space used within our banking database is split approximately 50/50 between table spaces and indexes.

For the SAP Account Settlement workload (batch processing), our measurements show that the total batch elapsed time was shortened by 6% and the database server ITR improved by 3% for Huffman compression versus the fixed length baseline. For the SAP Day Posting workload (OLTP), the average response times are in the sub-second range, and comparable for both encoding techniques, but the database server ITR was down by 4% for the Huffman compression versus the fixed length baseline. Thus, there was a range of performance impact, from a slight degradation to a slight improvement depending on the workload, for an 18% reduction in space usage.

A better compression ratio may reduce I/O and affect the performance of the application. The compression ratio depends on the characteristics of the data. The performance impact depends on the type of the workload, whether it is batch or online transaction processing—sequential or random data access pattern characteristics. The improvement seen in the SAP Account Settlement workload was due to the sequential data access pattern of the batch workload, which can benefit from compression if more records can be stored in a data page, resulting in reduced numbers of data page I/Os. However, the SAP Day Posting workload shows no performance benefit due to its more random data access pattern.

Conversion to Huffman compression may not trigger an immediate storage space savings in certain instances. For some extremely small tables, Huffman compression can use a few more data dictionary pages which can negate the database NPAGES savings. In addition, if a table space is allocated manually with significantly large PRIQTY and SECQTY attributes, then it may need to be resized in order to effectuate the storage space savings from the Huffman compression.

In summary, IBM Z effectively enhanced its compression technology to address the explosive data growth trend. It is estimated that 80% of the world's corporate data resides or originates on the mainframes [5]. This number also says that mainframes are at the heart of many large businesses in the world. The IBM Z platform is trusted by many large global enterprises to run their business operations because of its reliability, availability, and security, as well as for its capabilities to process large volumes of data effectively and securely.

8.0 References

1. IBM z14 Microprocessor Chip Set

   https://www.hotchips.org/wp-content/uploads/hc_archives/hc29/HC29.22-Tuesday-Pub/HC29.22.90-Server-Pub/HC29.22.910-Z14-processor-Jacobi-IBM.pdf

2. Using Huffman Compression to Compress Your Data

   https://www.ibm.com/support/knowledgecenter/SSEPEK_12.0.0/perf/src/tpc/db2z_compressdatahuffman.html

3. Database Administration Guide for SAP on IBM Db2 for z/OS

   https://help.sap.com/viewer/db2_administration_guide

4. Large Systems Performance Reference for IBM Z

   https://www-01.ibm.com/servers/resourcelink/lib03060.nsf/pages/lsprindex?OpenDocument

5. Storage: The Future Scape of IT

   https://www.ibm.com/blogs/systems/the-futurescape-of-it/

6. SAP on IBM Z Reference Architecture: SAP for Banking

   https://www.sap.com/documents/2015/07/8aab7888-5b7c-0010-82c7-eda71af511fa.html