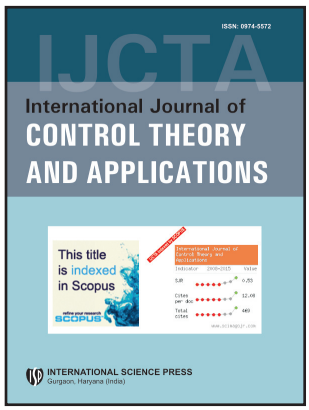

International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 24 • 2017

A New Approach to Increase the Storage Efficiency of Databases Using BIT Representation

Nimisha E.[a], Shyama P.[b] and Rajalakshmi V.R.[c]

[a,b]Student at Amrita Vishwa Vidhyapeetham, Department of Computer Science and IT, Kochi, India. Email: [a]nimishamohan.e@gmail.com; [b]shyamapradeep.93@gmail.com

[c]Faculty at Amrita Vishwa Vidhyapeetham, Departement of Computer Science and IT, Kochi, India. Email: rajiprithviraj@gmail.com

Abstract: Database is a collection of data such as schemas, tables, queries, reports, views and other objects. Typically, a computer database contains a group of data records or files. These records may be sales transactions, product catalogs and inventories, customer profiles etc. This information can be easily accessed, managed and updated. Some data may be repeating and are vital information that consumes a large area for data storage. So, compression of these data is necessary without losing any of them. Compression of data is a way of reducing the number of bits needed to store or transmit data. Compressions are of two types: Lossless or Lossy compression. In this paper, we are suggesting an algorithm which employs lossless compression in which the distinct, repetitive attribute values are represented as bits. According to the general rule; with n bits we can represent 2^n unique combinations. Also, the integrity of the data is preserved in lossless compression.

Keywords: Database, Database Management System, Data compression, Lossless compression, Compression algorithm.

1. INTRODUCTION

Database is defined as a systematic collection of information or data which can be easily accessed, updated and manipulated by the user. Database management system (DBMS) is a technology that creates and handles database on a computer. It is an aggregation of programs that enables the users or programmers to create, fetch, update and handle data in a database.

1.1. Types of Database

According to their structural or organizational approach the databases are classified as:

1. Distributed database: This type of database can be scattered among different point in a network. It is controlled by Distributed Database Management System (DDBMS).
2. Object oriented programming database: In this database, the data records and files are represented in the form of classes and objects. It is controlled by Object Database Management System (ODBMS). MySQL is a popular example of object oriented DBMS.

3. Relational database: In relational database, the data items are grouped into tables. The data can be retrieved or reassemble in many different ways without re-arranging the tables in the database. It is controlled by Relational Database Management System (RDBMS).

A relational database is a group of tables that contains rows and columns, in which each column represents an attribute or predefined category and each row gives the attribute values. The database entries are large in number that contains distinct and repetitive data which can either be vital or redundant. Such data require a large amount of storage space thus increasing the storage cost. So compression can be applied thereby increasing the storage efficiency and reducing the storage cost of the database.

1.2. Data Compression

Data compression: It is defined as a technique to lessen the number of bits required to transmit or store data. Compression helps in reducing the resource usage, such as transmission capacity or data storage to a great extend [1][13][14]. Also, exchange of these compressed files can be done easily over the internet since they upload and download much faster than the uncompressed files [8][15]. The ability of decompressing the original file at any time from the compressed one is a necessary factor. In other words, compression of data is a technique of converting data to smaller bits than the original representation so that it takes [8] only small area for storage and less time for transmission [8].

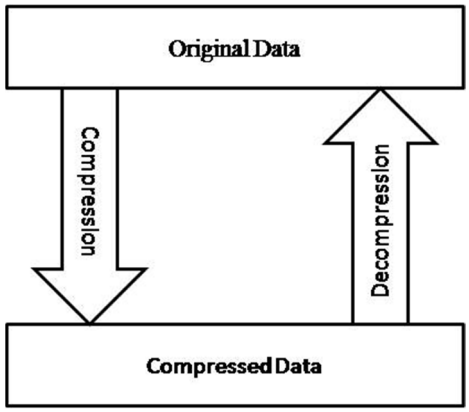

Figure 1: Data Compression and Decompression

Compression can be categorized into Lossy or Lossless [2][4][5][19].

Lossy Compression: In this compression technique the compressed data cannot be decompressed back to 100% of its original form [4]. Lossy methods provide high level of compression resulting in smaller compressed files [14].It may remove some portion of the original video or audio files. Examples are JPEG image, MPEG video and MP3 audio formats which are most commonly used nowadays.

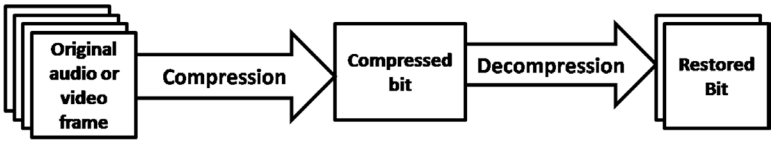

Figure 2: Lossy Compression

Lossless compression: In this technique the original data can be reconstructed from its compressed form without any loss [18]. Hence they are called as reversible compressions [7]. These techniques are used to compress vital information such as medical images and text legal data computer executable files etc [14].

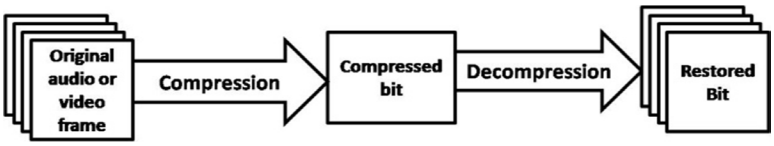

Figure 3: Lossless Compression

Lossless compression algorithm decreases the size of the file with no loss. When these files are saved it is compressed and when it is decompressed or opened, we can retrieve the original data [4]. There are various lossless compression algorithms that have been proposed so far but main techniques used are:

Run Length Encoding algorithm: Referred as RLE and is probably the easier lossless compression technique [13]. It uses count number and a single value to replace large sequences of repeating data values within a file [6]. Consider the string of data given below (16 bytes).

ABCCCCCCDEEEEEEF

Applying RLE the original string get compressed into 10 bytes and the compressed form look like: AB*6 C D *6E F

Here the successive series of symbols are identified as runs and the others as non runs [16]. These runs are used for compressing the original file [4]. Some type of redundancy is carried out by this algorithm. It examine for repeating symbols in a large sequence of the files. The major function of this algorithm is to Figure out the runs of the source file. Also it traces the length and symbol of each run. The RLE algorithm uses those runs for the compression processes and the non-runs are not using at any of the stages of compression [4].

Huffman Encoding Algorithm: [20]A lossless compression algorithm and it is also known as Huffman compression algorithm [20]. It is based on the frequentness of occurrence of a symbol in the file that is being compressed [6]. This algorithm is based on the statistical coding, which means that the probability of a symbol has a direct bearing on the length of its representation. The more probable the occurrence of a symbol is, the smaller will be its bit-size representation [10]. In many files, certain characters are used more than others. Using binary representation, the total number of bits needed to represent each character depends upon the number of characters that have to be depicted. Using one bit we can represent two characters that is, the first character can be substituted with 0 and the second one with 1. Similarly, with two bits we can represent four characters and so on [17]. Two kinds of Huffman Encoding Algorithm proposed are: Static Huffman Encoding and Adaptive Huffman algorithms.

Consider the following data: AAAAABBBBCCC [17]

Here the frequency of A is 5, B is 4 and C is 3.By representing each of these characters using a fixed code length of 2 bits, then the total number of bits required to store this data file would be (2x5)+(2x4)+(2x3) that is, 24 [17].

By Huffman Encoding Compression technique, the data can be compressed into smaller bits. The more frequently occurring characters would be represented by smaller bits: A by the binary code 0, B by the code 10, C by the code 11. Thus A, B, C requires only 1 bit, 2 bit and 2 bit respectively. Therefore, the total bits needed for the data file becomes (1x5)+(2x4)+(2x3) that is,19.In the above example, the final compressed file requires only less number of bits by assigning smaller binary codes to more frequently occurring characters[17] [4].

Arithmetic Encoding Algorithm: Another common Lossless compression method is the Arithmetic Encoding Algorithm. Here, in this compression method, the whole source message is represented by using a part or fraction instead of a codeword to symbolize the text symbols. The probability of occurrence and the cumulative frequency of symbols are taken into account and cumulative probability range is used in both the compression

and decompression techniques [4]. During the encoding process, the cumulative probability calculation and the range creation is done in the beginning itself. The source message is read character by character and while reading the source message, the corresponding range of the character within the cumulative probability range is selected. Then, the selected range is divided into sub parts according to the probabilities of the character. Similarly, the next character is read and the corresponding sub range is selected. In this way, the characters are read repeatedly until the end of the message is encountered. The number which taken from the final sub range is used as the output of the encoding process and the entire message is represented by this number, which will be a fraction [4].

Lempel Zev Welch Algorithm: Lempel Zev Welch compression algorithm or simply LZW is another kind of lossless compression algorithm [4]. Instead of a statistical model, here a dictionary based compression algorithms are used [4][2]. This compression is based on a dictionary. A dictionary is defined as a table like structure which contains a set of all possible word of a language and the indexes are used to represent the bigger and repeating words in a dictionary. The previously seen patterns are stored and indexed by this dictionary. In the process of compression, the index values assigned are substituted to the repeatedly occurring string patterns. For decompression, transmission of dictionary with the encoded message is not required as it is created dynamically in the compression process. Thus, this algorithm is also known as an Adaptive compression algorithm [4].

2. METHODOLOGY

2.1. PL/SQL

PL/SQL stands for procedural language which is an extension of SQL [11]. That is, it is a combination of SQL together with the procedural features of programming languages and it is designed specifically to hold SQL statements within its envelop.PL/SQL was developed in the early 90's by Oracle Corporation to improve the capabilities of SQL. PL/SQL consists of procedural language elements such as declaration, conditions, looping etc and also able to handle the runtime exceptions.

2.2. SQL Command Categories

Depending on their functionalities the SQL commands are divided into four major categories: DDL, DML, TCL and DCL.

- **Data Definition Language or DDL:** These commands are used for creating, modifying and dropping the structure of database objects. The DDL commands are: CREATE, ALTER, RENAME, DROP and TRUNCATE.
- **Data Manipulation Language or DML:** These commands are used for storing, modifying, retrieving. The DML commands are: INSERT, UPDATE, SELECT and DELETE.
- **Transaction Control Language or TCL:** These commands are used for handling the changes which are affecting the data. The TCL commands are: COMMIT, ROLLBACK.
- **Data Control Language or DCL:** These SQL command provides security to database objects. The DCL commands are: GRANT, REVOKE.

2.3. PL/SQL DYNAMIC SQL

Dynamic SQL: It is a programming approach in which the SQL statements are being created and processed at the runtime itself [12]. This kind of programming approach is useful in many ways:

1. While writing general-purpose, adaptable programs such as ad hoc query
2. For programs that must run DDL Statements
3. When SQL is not supported as static SQL
4. When the data types or number of its input and output variables or the complete text of a SQL statement are not known at the compilation time.

3. RELATED WORKS

Shrusti Porwal, Yashi Chaudhary, Jitendra Joshi, Manish Jain, "Data compression methodologies for lossless data and comparison between algorithms", International Journal of Engineering Science and Innovative Technology (IJESIT), Volume 2, Issue 2, March 2013.[1]Here the author compared the lossless data compressions and their performances [1]. The lossless compressions used in this paper were Huffman and arithmetic encoding [1]. Stepwise algorithmic processes and various performance measures had been performed according to the criteria, to analysis which technique is better. The measures used to calculate the performance were: compression ratio, compression speed, decompression speed, memory space needed, compressed pattern matching and permits random access. Finally, it is concluded that arithmetic encoding results the best compression ratio compared to Huffman compression. [1]

Amandeep Singh Sidhu, Er. Meenakshi Garg, "Research paper on text data compression algorithm using hybrid approach", International Journal of Computer Science and Mobile Computing(IJCSMC), Vol. 3,Issue.12, December 2014, pg.01-10.[8].Here, a hybrid way of compression is performed to compress the text data. A new and enhanced dynamic bit reduction algorithm based on lossless compression was developed to compress and decompress the text data. Various analysis techniques have been performed on different datasets such as special characters, numeral, alphanumeric, random etc. From the conclusion it is clear that the result analysis of this proposed system shows better result in terms of Compression ratio and saving percentage as compared to the existing techniques such as Bit Reduction and Huffman Coding. [8]

Haroon Altaraweh, Mohammed Altarawneh, "Data Compression Techniques on Text Files: A Comparison study", international journal of computer applications(0975-8887) volume 26-No 5, July 2011.[9] In this paper a study on different methods of data compression algorithms on English text files were carried out. The compression techniques were LZW, Huffman, Fixed-length code (FLC), and Huffman coding algorithm, and Fixed-length code (HFLC). They estimated and tested these algorithms on text files with different size. and the comparison is done in terms of compression Size, Ratio, speed and Entropy. Finally it is concluded that Lempel Zev Welch (LZW) is the best method compared to other methods tested [9].

Aarti, "Performance Analysis of Huffman Coding Algorithm", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 5, May 2013. [10] Huffman coding was proposed on text files as well as on various image files. The output of the experiment shows that the higher code redundancy helps to achieve more compression. And it was concluded that Huffman coding is an efficient method for image compression and decompression to some extend and the decoded output is similar and almost close to the original image used. Also, it states that the image compression depends on the number of pixels, size and compression ratio of the image used. By all these factors the author came to a conclusion as Huffman coding is a good technique for the compression of text data and general types of image files [10].

4. PROPOSED SYSTEM

4.1. Algorithm

Step 1: Select the repeated attributes from the selected column.

Example: Select count (DISTINCT product_category from superstoresales) as internalquery;

Step 2: Select the count of bits needed to represent the attribute (according to general rule, with n bits we can represent 2^n unique combinations).

Example: Select a = ceil(sqrt(b));

Step 3: Create a new table with the attributes and their corresponding bit values.

Example: Create table product_category(attributes VARCHAR(10), bitvalue NUMBER(2));

Step 4: Find the binary equivalent up to 'a' and insert the attribute and their corresponding bit value.

Example: insert into product_category(attributes, bitvalue)values('&attributes',&bitvalue);

Step 5: Update the original table with the corresponding bit values.

Example: Update superstoresales SET office supplies = 00, furniture = 11, technology = 01, WHERE province = Nunavut;

4.2. Data Manipulation Operations

The steps to be taken when data manipulation operations are performed:

INSERT: When new values are inserted, calculate the count of repeated values, recalculate the bit values and find the binary equivalent for the same.

DELETE: When new values are deleted, same process as insertion to be performed, that is calculating the count, recalculating the bit values and finding the binary equivalent for the current.

UPDATE: Updating the index table with the new value.

SELECT: We can select and check values from the index table instead of original table.

5. EXPERIMENTS AND RESULTS

A sample dataset of Superstore Sales is being used for the experimentation of this paper, which contains distinct repetitive values. The table structure is as follows:

Customer Na	Provinc	Regio	ustomer Seg	roduct Cate	Product Sub-Categ	Product Name
Muhammed Macl	Nunavut	Nunavut	Small Business	Office Supplies	Storage & Organization	Eldon Base for stackable storage shelf, platinum
Barry French	Nunavut	Nunavut	Consumer	Office Supplies	Appliances	1.7 Cubic Foot Compact "Cube" Office Refrigerators
Barry French	Nunavut	Nunavut	Consumer	Office Supplies	Binders and Binder Acces	Cardinal Slant-D® Ring Binder, Heavy Gauge Vinyl
Clay Rozendal	Nunavut	Nunavut	Corporate	Technology	Telephones and Commur	R380
Carlos Soltero	Nunavut	Nunavut	Consumer	Office Supplies	Appliances	Holmes HEPA Air Purifier
Carlos Soltero	Nunavut	Nunavut	Consumer	Furniture	Office Furnishings	G.E. Longer-Life Indoor Recessed Floodlight Bulbs
Carl Jackson	Nunavut	Nunavut	Corporate	Office Supplies	Binders and Binder Acces	Angle-D Binders with Locking Rings, Label Holders
Carl Jackson	Nunavut	Nunavut	Corporate	Office Supplies	Storage & Organization	SAFCO Mobile Desk Side File, Wire Frame
Monica Federle	Nunavut	Nunavut	Corporate	Office Supplies	Storage & Organization	SAFCO Commercial Wire Shelving, Black
Dorothy Badders	Nunavut	Nunavut	Home Office	Office Supplies	Paper	Xerox 198
Neola Schneider	Nunavut	Nunavut	Home Office	Office Supplies	Paper	Xerox 1980
Neola Schneider	Nunavut	Nunavut	Home Office	Office Supplies	Rubber Bands	Advantus Map Pennant Flags and Round Head Tacks
Carlos Daly	Nunavut	Nunavut	Home Office	Office Supplies	Appliances	Holmes HEPA Air Purifier
Carlos Daly	Nunavut	Nunavut	Home Office	Technology	Computer Peripherals	DS/HD IBM Formatted Diskettes, 200/Pack - Staples
Claudia Miner	Nunavut	Nunavut	Small Business	Office Supplies	Binders and Binder Acces	Wilson Jones 1" Hanging DublLock® Ring Binders
Neola Schneider	Nunavut	Nunavut	Home Office	Furniture	Office Furnishings	Ultra Commercial Grade Dual Valve Door Closer
Allen Rosenblatt	Nunavut	Nunavut	Small Business	Office Supplies	Envelopes	#10-4 1/8" x 9 1/2" Premium Diagonal Seam Envelopes
Sylvia Foulston	Nunavut	Nunavut	Home Office	Furniture	Bookcases	Hon 4-Shelf Metal Bookcases
Sylvia Foulston	Nunavut	Nunavut	Home Office	Furniture	Tables	Lesro Sheffield Collection Coffee Table, End Table, Cent
Jim Radford	Nunavut	Nunavut	Corporate	Technology	Telephones and Commur	g520
Jim Radford	Nunavut	Nunavut	Corporate	Technology	Telephones and Commur	LX 788
Carlos Soltero	Nunavut	Nunavut	Consumer	Office Supplies	Labels	Avery 52
Carlos Soltero	Nunavut	Nunavut	Consumer	Office Supplies	Rubber Bands	Plymouth Boxed Rubber Bands by Plymouth
Carl Ludwig	Nunavut	Nunavut	Corporate	Office Supplies	Binders and Binder Acces	GBC Pre-Punched Binding Paper, Plastic, White, 8-1/2"

Figure 4: Dataset: Superstore Sales

From the above table we have selected the columns which have distinct repetitive values which needs large storage space. For increasing the storage efficiency, we are applying this new approach. Firstly, we are calculating the counts of distinct repetitive values from each selected column.

For example, the count of repetitive values of the column "Product Category" is 3. According to the general rule, with n bits we can represent 2^n unique combination. Here we have 3 unique combinations and we need two bits to represent. The bit values for these combinations are: 00, 11, 01, 10.

Now we are substituting these values to the attributes and creating an index table with the bit value and their corresponding attribute value.

Table 1
Index table

Product Category	
Bit value	*Attribute value*
00	Office supplies
11	Furniture
01	Technology
10	-

Likewise, we are creating index tables for each selected column with the bit value and the corresponding attribute. After creating the index tables for all columns, we have to update the original table with the bit values assigned. The updated table is as below:

Provin	Region	Customer Segment	Product Category	Product Sub-Categor	Product Name
00	00	00	11	0001	Eldon Base for stackable storage shelf, platinum
00	00	11	11	1110	1.7 Cubic Foot Compact "Cube" Office Refrigerators
00	00	11	11	0100	Cardinal Slant-D® Ring Binder, Heavy Gauge Vinyl
00	00	00	00	1100	R380
00	00	11	11	1110	Holmes HEPA Air Purifier
00	00	11	01	1010	G.E. Longer-Life Indoor Recessed Floodlight Bulbs
00	00	00	11	0100	Angle-D Binders with Locking Rings, Label Holders
00	00	00	11	1000	SAFCO Mobile Desk Side File, Wire Frame
00	00	00	11	1000	SAFCO Commercial Wire Shelving, Black
00	00	01	11	0011	Xerox 198
00	00	01	11	0011	Xerox 1980
00	00	01	11	0011	Advantus Map Pennant Flags and Round Head Tacks
00	00	01	11	1110	Holmes HEPA Air Purifier
00	00	01	00	0001	DS/HD IBM Formatted Diskettes, 200/Pack - Staples
00	00	10	11	0100	Wilson Jones 1" Hanging DublLock® Ring Binders
00	00	01	01	1010	Ultra Commercial Grade Dual Valve Door Closer
00	00	10	11	1111	#10-4 1/8" x 9 1/2" Premium Diagonal Seam Envelopes
00	00	01	01	0011	Hon 4-Shelf Metal Bookcases
00	00	01	01	0011	Lesro Sheffield Collection Coffee Table, End Table, Center
00	00	11	00	1100	g520
00	00	11	00	1100	LX 788
00	00	00	11	1111	Avery 52
00	00	00	11	1111	Plymouth Boxed Rubber Bands by Plymouth
00	00	11	11	0100	GBC Pre-Punched Binding Paper, Plastic, White, 8-1/2" x 1

Figure 5: Dataset updated with Bit value (Compressed dataset)

In the original table, column "Product Category" requires 28 bits for the attribute Office supplies, 18 bits for Furniture, and 20 bits for Technology. That is, in the original table total of 66 bits are using repeatedly.

In the updated table, only 2 bits for the attribute office supplies, 2 bits for furniture and 2 bits for Technology. Total of 6 bits are using repeatedly in this updated (compressed) table. That is, if there are x number of attributes, the original table requires (66xX) number of bits and the updated table requires (6xX) number of bits.

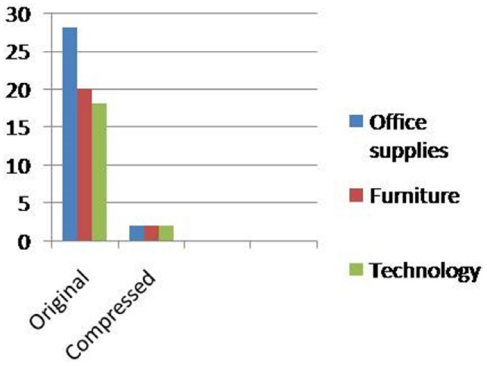

Figure 6: Bar chart representing the storage space needed for the attributes in Product category

From the bar chart, it is clear that the compressed table requires only small number of bits to store data compared to the original table. Here, values in the table are compressed without any loss and storage space of the dataset is increased. Thus, this compression algorithm helps to increase the storage efficiency.

5.1 Advantages

- No loss of data
- Storage efficiency is increased.
- Normalization is not needed.
- Applicable to any type of data such as characters, numbers, images etc.
- Secure: as it displays only the bit values in the original table.

6. CONCLUSION

In this proposed work, a new way to increase the storage capability of dataset is developed by compressing the dataset into BIT values. Various analysis techniques have been carried out on different datasets. The datasets with large distinct and repetitive data are compressed into bit values and the compressed dataset is compared with the original dataset. By considering the bits needed, compressed one is more storage efficient as compared to the original dataset.

From the result, we can conclude that this proposed system provides a good compression results in terms of storage efficiency and is a kind of lossless compression technique in which no loss of data when it is compressed.

7. FUTURE ENHANCEMENT

For more security, encryption can also be carried out along with the compression. So that no one can alter the dataset or make changes.

REFERENCES

[1] Shrusti Porwal, Yashi Chaudhary, Jitendra Joshi, Manish Jain,Data "Compression Methodologies for Lossless Data and Comparison between Algorithms", International Journal of Engineering Science and Innovative Technology(IJESIT),Volume 2, Issue 2, March 2013

[2] Nishad P M, R.Manicka Che zian, "Enhanced LZW (Lempel-Ziv-Welch) Algorithm by Binary Search with Multiple Dictionary to Reduce Time Complexity for Dictionary Creation in Encoding and Decoding", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 3, March 2012

[3] Paul G. Howard, Jerey Scott Vitter, "Practical Implementations of Arithmetic Coding", A shortened version appears in the proceedings of the International Conference on Advances in Communication and Control (COMCON3),Victoria, British, Columbia, Canada, October16-18,1991

[4] S.R. KODITUWAKKU,U, S.AMARASINGHE, "COMPARISON OF LOSSLESS DATA COMPRESSION ALGORITHMS FOR TEXT DATA", S.R. Kodituwakku et. al.,/Indian Journal of Computer Science and Engineering, Vol. 1 No 4 416-425

[5] M.Sundaresan,E.Devika, "Lossy and Lossless Compression using various Algorithms", International Journal of Computer Applications (0975 – 8887) Volume 65– No.20, March 2013

[6] Eman Abdelfattah and Asif Mohiuddin, "PERFORMANCE ANALYSIS OF MULTIMEDIA COMPRESSION ALGORITHMS", International journal of computer science & information Technology (IJCSIT) Vol.2, No.5, October 2010

[7] Dr.E.KANNAN, G. Murugan, "Lossless Image Compression Algorithm For Transmitting Over Low Bandwidth Line ", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 2, February 2012, ISSN: 2277 128X

[8] Amandeep Singh Sidhu, Er. Meenakshi Garg, "Research Paper on Text Data Compression Algorithm using Hybrid Approach", International Journal of Computer Science and Mobile Computing (IJCSMC), Vol. 3, Issue.12, December 2014, pg.01 – 10

[9] Haroon Altarawneh, Mohammad Altarawneh, "Data Compression Techniques on Text Files: A Comparison Study", International Journal of Computer Applications (0975 –8887) Volume 26–No.5, July 2011

[10] Aarti, "Performance Analysis of Huffman Coding Algorithm", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 5, May 2013

[11] Hitesh KUMAR SHARMA, Ranjit BISWAS, Aditya SHASTRI, "PL/SQL and Bind Variable: the two ways to increase the efficiency of Network Databases", Database Systems Journal Vol. II, No. 4/20119

[12] VAMSI KRISHNA MYALAPALLI, "Design of a model to tune the PL/SQL program and improve the performance of DBMS", International Journal of Advanced Computer Communications and Control, Vol. 02, No. 01, January 2014I, JACCC 020103

[13] I Made Agus Dwi Suarjaya, "A New Algorithm for Data Compression Optimization", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No.8, 2012

[14] RupinderSingh Brar, Bikramjeet singh, "A Survey on Different Compression Techniques and Bit Reduction Algorithm for Compression of Text/Lossless Data", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 3, March 201

[15] Manjeet Kaur, Er. Upasna Garg, "A Review of Various Data Compression Techniques to form a New Technique for Text Data Compression", Imperial Journal of Interdisciplinary Research (IJIR), Vol-1, Issue.5,2015.

[16] M.Ravi Kumar,S.Manoj Kumar, "Multi Benfit's Through Compression For Large Data Stored In Cloud (An integrated advantages of data compression in cloud)" International Journal Of Engineering And Computer Science IJECS Volume 2 Issue 4 April,2013 Page No. 991-996.

[17] Mrs. Bhumika Gupta, "Study Of Various Lossless Image Compression Technique", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS),Volume 2, Issue 4, July –August 2013

[18] Anmol Jyot Maan, "Analysis and Comparison of Algorithms for Lossless Data Compression", International Journal of Information and Computation Technology.ISSN 0974-2239 Volume 3, Number 3 (2013), pp. 139-146

[19] Gaurav Vijayvargiya, Dr. Sanjay Silakari, Dr.Rajeev Pandey, "A Survey: Various Techniques of Image Compression", (IJCSIS) International Journal of Computer Science and Information Security, Vol. 11, No. 10, October 2013".

[20] JAGADISH H. PUJAR, LOHIT M. KADLASKAR, "A NEW LOSSLESS METHOD OF IMAGE COMPRESSION AND DECOMPRESSION USING HUFFMAN CODING TECHNIQUES", Journal of Theoretical and Applied Information Technology.