International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

Compression Techniques: Key to Effective Data Transmission

Dhanashree Toradmalle, Jayabhaskar Muthukuru, B Sathyanarayana

Abstract: ***In the digital world today, data is growing tremendously. The onus lies on the network to compute, process, transfer and store this data. There is a direct proportional relationship between the size of data to the efficiency of a given system. The major challenge that systems face today is the size of data thereby the vision of these systems is to compress the data to its maximum so that the storage space, processing time is reduced thereby making the system effective. As DC ideas result to viable usage of accessible storage space and effective transfer speed, various methodologies were created in a few angles. An itemized overview on many existing DC strategies is expressed to address the present necessities in lieu of the information quality, coding plans and applications in request to break down how DC strategies and its applications have advanced a similar investigation is performed to recognize the commitment of inspected procedures as far as their qualities, fundamental ideas, exploratory variables and constraints.***

Keywords: Data compression, decompression, coding, encoding, decoding

I. INTRODUCTION

One cannot deny the catastrophic contribution of the compression technology in today's high-end applications over the internet. The massive storage, bandwidth and processing time requirements of systems are reduced thereby reducing the cost and response time over the internet. An encoding calculation that takes a message and creates a "compacted" portrayal (ideally with less bits), and a deciphering calculation that remakes the first message or some estimate of it from the packed portrayal are the essential two phases [1] of any compression technique These two parts are commonly unpredictably integrated since the two of them need to comprehend the common compacted portrayal.

Definition [2]: Data compression is the process of minimizing the amount of data required to represent a given quantity of information.

Manuscript published on 30 September 2019
* Correspondence Author
Dhanashree Toradmalle, Department of CSE, KLEF, India
Email: dhanashree.kt@gmail.com
Jayabhaskar Muthukuru, Department of CSE, KLEF, India Email: jayabhaskarm@gmail.com
B Sathyanarayana Department of Computer Science & IT, Sri Krishnadevaraya University, India Email:bachalasatya@yahoo.com

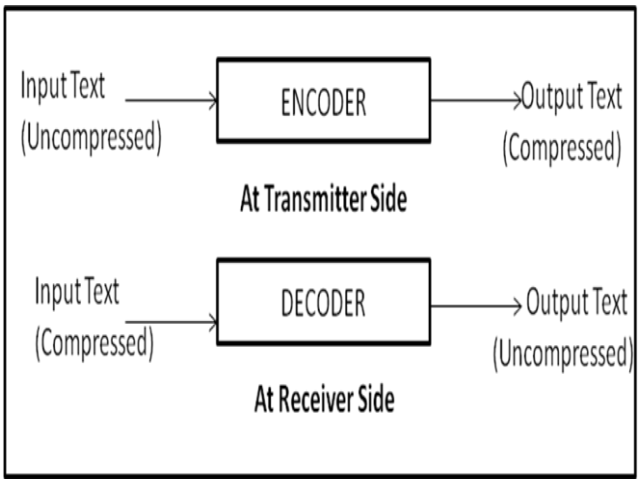

Figure.1 Coding-Decoding Process

A. Compression Quality Metrics

The metrics based on which the compression methods can be evaluated are based on the nature of the applications. The principle concerns for Data compression methods are time effectiveness and space effectiveness. The quantity of redundancy in the given data determines the compression behavior of the algorithm. Depending on the compression behavior i.e whether we allow the reconstruction of the data to be identical to the source data we classify data compression methods broadly into two major types [3]:

- Lossy compression technique: A lossy compression [4] algorithm deals with losing some amount of data in the encoding process thereby reducing the size of the original message.

- Lossless data compression technique: None of the data elements are at loss in lossless data compression algorithm [5] during the process of encoding the file, thereby assuring to reproduce exactly same data as at the input. The lossless data compression algorithms should be encouraged if data loss is not desirable

B. Metrics used in Lossy Compression Techniques

How well is the data size reduced in comparison to the original size highlights the efficacy of the Lossy compression technique. The quality metrics for the Lossy methods are detailed below:

Compression Ratio: This is the proportion of the size data after compression to size of data before compression.

$$\text{Compression Ratio} = \frac{\text{Size of data after applying Compression}}{\text{Size of data before applying Compression}}$$

Compression Factor: This is the inverse of Compression Ratio. It is proportion of the size data before compression to size of data after compression.

$$\text{Compression Factor} = \frac{\text{Size of data before applying Compression}}{\text{Size of data after applying Compression}}$$

Saving Percentage: Reduction of the data after compression represented in percentage.

$$\text{Saving Percentage} = \frac{\text{Size of data before applying Compression} - \text{Size of data after applying Compression}}{\text{Size of data before applying Compression}}$$

Compression Time [6]**:** Time taken for compression and decompression must be considered independently. If the time taken to compress and decompress are acceptable i.e. within the acceptable limit of the given application the method is said to be acceptable with respect to the compression time quality metric.

II. THEORY

Lossless Data Compression Algorithms

Coding is an integral part of any data compression algorithm which reflects the performance of the algorithm. The following section is an attempt to study the coding methods and analyze them independently.

A. Huffman Coding:

Huffman [7] in 1952 presented the acclaimed coding method which can compress practically all types of documents. It is a variable length coding used in the lossless method of compression. The technique works on the concept of assigning codes to the input characters depending on their frequency of occurrence. The more frequent the character appears in the text the shorter is the code attached to the character reducing the number of bits needed to represent it. The method is drawn with the assistance of a tree like structure known as the Huffman tree which helps is traversing forward and backward the in the encoding and decoding phases. It is interestingly decodable and comprises of two parts, for example, building Huffman tree from information arrangement and navigating the tree to dole out codes to characters. Huffman coding is the most popular methods used due to its simplicity and time effectiveness. Researchers have come up with several variations [8] [9] to the basic Huffman Coding method to cash on its efficacy.

- *Adaptive Huffman coding:*

  The areas of concern of processing with Huffman codes are

  - It is a static code
  - The algorithm views all the data to be compressed beforehand leading to threats to security.
  - It needs a self-assertive measure of memory available to the user for the formation of the Huffman trees and their processing.
  - The decoding phase should be well trained with the codebook used by the encoding phase.

  All things considered, in some cases a few applications one may need to encode data originating from a large letter set. While Huffman algorithm uses memory relative to the number of letters Adaptive Huffman [10] uses memory relative to the quantity of recurrence classes which is, consistently low or equivalent to the size of the letter set. Steven Pigeon [11] describes a method of adaptive Huffman coding which uses sets of symbols rather than single symbols represented at the leaf of the tree thereby generating the symbols having same frequency at the leaf nodes. Set migration and rebalancing are used effectively to utilize the memory optimally.
- *Length limited/ Minimum variance Huffman coding* [12]: The objective to accomplish a base weighted way length, bearing in mind the limitation that the length of each codeword must be not exactly as the given constant.
- *Canonical Huffman Coding* [13][14]: The bit lengths of the standard Huffman codes generated for each symbol is used. The Canonical Huffman Encoding process includes processes wherein the symbols are filtered and sorted, and further used to build a Huffman tree. Unlike in Huffman Coding where the entire tree is passed to the decoder, the encoding is done such that only the length of the symbols in the tree is required by the decoder.
- *Golomb code* [15][16]: The Golomb-Rice codes have a place with a group of codes intended to encode whole numbers with the supposition that the bigger a number, the lower its likelihood of event. It selects optimum code parameters amongst the permittable values which further give minimum code length to encode the given sequence of sample.
- *Tunstall code* [17]: Tunstall codes are counterparts to Huffman codes for variable to block coding.

B. ***Arithmetic Coding*** [18][19]:

Arithmetic coding is a lossless entropy-based coding method. The message is represented using fractional numbers. The intervals required to represent the message reduces as this interval becomes longer. The more likely symbols reduce the range by less than the unlikely symbols and hence add fewer bits to the message.

- *Adaptive Arithmetic Coding* [20]:
  The feature of Adaptability of Arithmetic coding, that is responsiveness to change in frequency tables during processing marks an edge over other compression techniques. Until the frequency tables in decoding are reassigned in similar fashion as of encoding the decoded data will not match the original data.
- *Binary Arithmetic Coding*:
  This method is accepted universally as data symbols can be represented as binary codes. The tradeoff of the method is that though it simplifies coding each binary data symbol, its final throughput of information (actual information bits) cannot be larger than one bit per coded symbol, which normally means one bit per few CPU clock cycles.

C. ***LZ Coding*** [21]

This is a dictionary-based method, wherein a dictionary is created which contains single-character strings which are relative to the possible input characters. The encoding algorithm scans the dictionary thoroughly until it finds the string in the dictionary. If the substring is not found the new string is added to the existing dictionary. In the decoding phase the values of encoded set are read, and a dictionary is generated in similar fashion like the encoding phase. The method proves efficient in cases of highly repeated patterns of text. LZ has many variations proposed [22] based on the construction of dictionaries viz; LZ77, LZ78, LZW

D. ***RLE*** [23]:

Run Length Encoding is a Lossless technique which is proactively used in applications when the constants in the symbols are highly redundant.

E. ***Burrows and Wheeler*** [24]:

BWT uses the method of lexicographical reversible permutation of the characters of a string. It primarily builds an array which has long sequences of identical characters clustered together with an explicit ordering which enables the algorithm to compress better. The remarkable thing about BWT is that this method is reversible with minimal data overhead leading to lower costs of compression.

III. DISCUSSION

Table.1 Lossless Coding Techniques

Coding Method	Feature	Variants	Merits	Applications
Huffman Coding	Entropy based	Adaptive Huffman coding Minimum variance Canonical coding Golomb coding Tunstall coding	Effective in all file formats	ZIP, ARG, JPEG, MPEG, PKZIP
Arithmetic Coding	Entropy based	Adaptive	Flexibility	JPEG, multimedia
		Binary		Applications
LZ Coding	Dictionary Based	LZ77 LZ78 LZW	Compress all kinds of data	TIFF, GIF, PDF, Gzip, ZIP, V.42,
RLE	High Redundant bits	-	Faster	TIFF, BMP, PDF and Fax

Table 1. discusses the Lossless Compression Techniques with their details briefing the readers with their scope of applications.

IV. CONCLUSION

Data Compression plays an inevitable role in data transmission and processing today. The author has made an attempt to discuss the data compression lossless techniques in view of presenting them and their variations and their scope of applications to guide the readers in selecting an appropriate method that best suits their requirement. No single technique is best suited to the variety of applications available in the digital world today. The selection of the coding technique is dependent on several parameters from application point of view. At few places a combination of more than a single technique may also prove beneficial

REFERENCES

1. Khalid Sayood, Introduction to Data Compression. Elsevier, 2012
2. Pu, I.M., 2006, *Fundamental Data Compression*, Elsevier, Britain.
3. David Salomon, Data Compression: The Complete Reference.Springer-Verlag, London, 2007
4. Mark Nelson, The Data Compression Book, M&T Press, 1991
5. Komal Sharma , Kunal Gupta "Lossless Data Compression Technique With Encryption Based Approach", International Conference on Computing, Communication and Networking Technologies (ICCCNT) 2017.
6. U. S.Amarasinghe, S.R. Kodituwakku "Comparison of lossless data compression algorithms for text data" Indian Journal of Computer Science and Engineering, Vol 1 No 4 pp 416-425
7. D.A. Huffman, "A method for the construction of minimum redundancy codes" , Proceedings of the I.R.E., v40 (1951) p 1098-1101
8. Y. Choueka , S.T. Klein , Y. Perl, "Efficient Variants of Huffman Codes in High Level Languages", Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval ACM June 1985
9. Lawrence. Larmore, Daniel S. Hirschberg, "A Fast Algorithm for Optimal Length-Limited Huffman Codes", Journal of the Association for Computing Machinery, Vol. 37, No. 3, July 1990, pp. 464-473
10. Dr. Muhammad Younus Javed , Mr. Abid Nadeem, "Data compression through Adaptive huffman coding scheme", 2000 IEEE
11. Stevcn Pigeon, Yoshua Bengio, "A memory efficient Adaptive Huffman Algorithm for very large sets of symbols", Proceedings DCC '98 Data Compression Conference
12. Sandeep G. S, Sunil Kumar B. S, D J Deepak, "An Efficient Lossless Compression Using Double Huffman Minimum Variance Encoding Technique", 2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)
13. Janarbek Matai_, Joo-Young Kimy, and Ryan Kastner, "Energy Efficient Canonical Huffman Encoding",
14. Shree Ram Khaitu, Sanjeeb Prasad Panday, "Canonical Huffman Coding for Image Compression" 3rd International Conference on Computing, Communication and Security (ICCCS), 2018 IEEE

15. Ryosuke Sugiura , Yutaka Kamamoto, Noboru Harada,Takehiro Moriya, “Optimal Golomb-Rice Code Extension for Lossless Coding of Low-Entropy Exponentially Distributed Sources”, IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 64, NO. 4, APRIL 2018
16. S.Domnic, “A New Method for Golomb-Rice parameter estimation”, 2017 IEEE
17. Francesco Fabris “On the Composition of Tunstall Messages”, IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 45, NO. 5, JULY 1999
18. Glen G. Langdon, Jr. And Jorma J. Rissanen , “A Double-Adaptive File Compression Algorithm”, IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. COM-31, NO. 11, NOVEMBER 1983
19. Ian H. Witten, Radford M. Neal, John G, “Cleary, arithmetic coding for Data compression”, Communications of the ACM: Volume 30 Issue 6, June 1987
20. Jing Wang*, Xuan Ji, Shenghui Zhao, Xiang Xie, Jingming Kuang, “Context-based adaptive arithmetic coding in time and frequency domain for the lossless compression of audio coding parameters at variable rate”, Journal on Audio, Speech, and Music Processing 2013, Springer.
21. Suzanne Bunton, Gaetano Borriello, “Practical Dictionary Management for Hardware Data Compression”, COMMUNICATIONS OF THE ACM/January 1992/Vo1,35, No.1
22. Kewen Liao, Matthias Petri, “Effective Construction of Relative Lempel-Ziv Dictionaries” WWW '16: Proceedings of the 25th International Conference on World Wide Web,April 2016
23. Capon, J. “A probabilistic model for run-length coding of pictures.” 1959IRE Trans. Inf. Theory 100, 157–163.
24. Burrows, M., Wheeler, D., “. A block-sorting lossless data compression algorithm.” Algorithm, Data Compression 18,1994

AUTHORS PROFILE

Ms.Dhanashree K Toradmalle is working as an Associate Professor in Shah & Anchor Kutchhi Engineering College, Mumbai. She is currently pursuing her PhD in Computer Science Engineering from K L E Foundation (Deemed to be University), Guntur, Andhra Pradesh in Computer Science Engineering. Her research areas include Computer Networks and Security.

Dr. M Jaya Bhaskar has 7+ years of industry and 6+ years of teaching experience and has interests in real time issues in Networks which lead to research in Network and Data Security and further implementation of different security techniques like cryptography and signcryption. He completed his PhD in Elliptical Curve Cryptography Implementation Approaches for Efficient Smart Card Processing from Sri Krishnadevaraya University, Ananthpuram in 2013.He is currently working as Associate Professor in K L E Foundation (Deemed to be University), Guntur, Andhra Pradesh. His research areas include Network and Information Security

Prof. B. Sathyanarayana received his Master of Computer Applications from Madurai Kamaraj University in 1988. He did his Ph.D in Computer Networks from Sri Krishnadevaraya University, Ananthpuram, A.P. India. He has around 30+ years of teaching experience. His Current Research Interest includes Computer Networks, Network Security and Intrusion Detection. He has guided many research scholars for PhD.He has published more than 50 research papers in National and International journals