Volume 3, Issue 11, November 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Technique of Data Analysis and File Compression Using Huffman Algorithm

C. Barath Kumar*
M.C.A
SITE, VIT University
Vellore, Tamilnadu, India

M. Varun Kumar
Assistant Professor
SITE, VIT University
Vellore, Tamilnadu, India

T. Gayathri
M.S (Software Engg)
SITE, VIT University
Vellore, Tamilnadu, India

N. Vanisri, S. Rajesh Kumar
M.C.A
SITE, VIT University
Vellore, Tamilnadu, India

Abstract: - The domain "Sun Zip" will reduce the overall number of bits and bytes in a file. So, the files can be easily transmitted over slow internet connection and it takes a less space in disk. Domain sun zip is based on software. The software is done by java. It can use in the system as a utility. The type of compression we use here is lossless compression. The user doesn't need to depend upon the third part software like WinZip, WinRAR, 7zip, etc. The aim of the system is to develop a system of improved facility. The system can overcome all the limitations of the previous system. Our software provides data accuracy and save disk space. The previous system has more disadvantage and difficulties to work well. This new system tries to eliminate or reduce those difficulties up to some extent. The main algorithm we used to create this domain is Huffman algorithm

Keyword: - Huffman Compression, Gzip Compression, Sun Zip.

I. INTRODUCTION

The Lossless compression means, it won't delete the File content to reduce the file size. The original file remains as it is, after it gets compressed also. For implementing this software, we are using Huffman algorithm. The Huffman encoding algorithm is an optimal compression algorithm when only the frequency of individual letter is used to compress the data. The idea behind the algorithm is that if you have some letters that are more frequent than others, it makes sense to use fewer bits to encode those letters than to encode the less frequent letters. Thus size of files can be reduced efficiently.

Huffman coding uses a specific method for choosing the representation for symbol, resulting in prefix code it is also called as "prefix-free codes" that express the common character using shorter strings of bits than are used for less common source symbol. Huffman was able to design in the most efficient compression method this type: no other mapping of individual source symbols to unique string of bits will produce a small average output size when the actual symbol frequencies agree with those used to create the code. A method was later found to do this in linear time if input probabilities. For a set of symbol with a uniformly probability distribution a number of members which is a power of two, Huffman coding is equivalent to simple binary block encoding. Huffman coding is such a widespread method for creating prefix codes that the term "Huffman code" is widely used as a synonym for "prefix-code".

II. EXISTING SYSTEM

The main problem in the existing system is the used more depend on the third party software's like WinZip, WinRAR, 7Zip etc., The existing system takes more time for compressing the files and the complexity involved in the process are very high and the other disadvantages are lack of data security, deficiency of data accuracy, time consuming. To avoid all these limitations and working more efficient, system needs to be computerized.

Major Drawbacks of the Existing System

A. Lack of Data Security

The major task of the software is to ensure that the data compressed must meet the satisfaction of reduced size of file, data must be secured. The system software doesn't have proper protection towards the data. The data can be protected by providing the password but those passwords can be easily broken and the data can be misused.

B. Deficiency of Data Security

A number of deficiencies exit in the system. Analysis the existing system there are many data accuracy issues takes place the major issues is the data often gets corrupted this is due to the compression problem. While compressing the data its frequency level is not reduced to the proper level that leads to the corruption of data.

C. Time Consuming: Most the software takes a large time to compress the files. Computer files are coded in such a way that spaces are left between coded segments. In addition, the same redundant information is repeated throughout the document. This redundancy is removed by file compression. Placing a reference back to the repeated information, it removes the letter. Thus data is converted to very efficient coding. It requires less storage space from an inefficient one. In computer language this is nothing but representation of data in an algorithmic form that uses less bits and bytes.

D. Dependency of Third Party Software's

There are many third party software's available in market. Even they can be classified into two types open source software and paid versions are available. Some of the software's are NCH Express Zip, WinZip, and WinRAR etc.

E. File Extension

Most of the third party software have their own extension like zip, rar, 7z once the file has been compressed it automatically changes into specific extension hence surly we must need specific software to view/recover the data.

III. SOLUTION METHODOLOGY

The aim of the system is to develop a system of improved facilities. The system can overcome all the limitations of the previous system. The system provides data accuracy and save disk space. The previous system has several disadvantage and many more difficult to work well. This new system tries to eliminate or reduce the difficulties up to some extent. This file compression or decompression is based on Huffman algorithm. The system helps the user to consume time. The system helps the users to work user friendly and they can easily do file compression without much time lagging. The system is very easy to design and to implement.

The system requires very low system resources and the system almost work in all configuration the system has got the following features

I. Ensure data accuracy and save sick space
II. Minimum time required for file compression
III. Greater efficiency and better service
IV. Easy to send files via E-mail
V. The user doesn't need to depend on any third party software
VI. File extension is optional

IV. RESULTS AND DISCUSSION

A. Compressing a File

In this module, we have to give the file while we need to compress, after the file is selected in the source path automatically the file name with the default extension of the algorithm will be displayed in Destination path. Finally the file will compressed with minimum file size.

B. Decompressing a File

It is the reverse process of compressing a file. First we have to select the compressed file, and then the file with its original extension (ex: txt, pdf, ppt, wmv, avi etc.) will displayed on the Destination Path. Finally the file will get decompressed with its original file size without any damage.

C. File Extension

It is a special process, here the user can use the default extension else it is possible to set their own extension for the file. It won't affect the original file extension.

Default extension for Huffman algorithm is (.huf)

Syntax: <File name><File Extension><Huffman Extension>
Example: java.pdf.huf

Default extension for the Gzip algorithm is (.sfe)

Syntax: <File name><File Extension><Gzip Extension>
Example: java.pdf.sfe

D. Without Extension

It is an additional feature, here the user can also ignore the extension link (used to set extension) from the coding and can compress the file. If they compress, the original file compressed with the same file extension but the file size will get reduced. No data loss will occur.

Example: Input: java.pdf
Output: java.pdf

V. CONCLUSION

The project sun zip is satisfying the required design specification. The system provides a user-friendly interface. Finally after Compressing ratio can be enhanced further more. Compressing multiple file and folders at a time. Changing the style of the icon. Increasing the data security to protect the data by using voice recognition. Enable the users to view files

that already compressed/decompressed using the third party software. The software is developed with a modular approach. Almost All the modules in the software have been tested and work successfully. Thus the system has fulfilled all the objectives identified and is able to replace the existing system. The constraints are met and over come successfully.

REFERENCES

[1] Peter Rauschert, Yuri Klimets, Jorg Velten and Anton Kummert. Very Fast Gzip Compression by means of content addressable memories, Faculty of Electrical, Information and Media Engineering, University of Wuppertal, Germany.

[2] Faisel Saeed, Huizhu Lu, G.E. Hedrick. Data Compression with Huffman Coding: An Efficient dynamic implementation using File partitioning, Department of Computer Science, Oklahoma State University.

[3] Yuan Jing. The Combinational Application of LZSS and LZW Algorithms for compression based on Huffman, Radio and TV university, Changzhou, China.

[4] Ren Weizheng, Wang Haobo, Xu Lianming, Cui Yansong (2011). Research on a Quasi-Lossless Compression Algorithm based on Huffman Algorithm, School of Electronic Engineering, Beijing, China.

[5] Xrysovalantis Kavousianos (2007). Multilevel Huffman coding: An Efficient Test data compression method for IP cores, Member, IEEE, and Emmanouil Kalligeros.

[6] Almut Herzog & Nahid Shahmeri (2005). An evaluation of java application containers according to security requirement.

[7] Mary Campione, Kathy Walrath & Alison Huml (2005).The Java tutorial continued published by Addison Wesley.

[8] Bruce Eckel (2006, February 10). Thinking in Java – 4th edition published by Addison Wesley.

[9] Peter Norton (1996). Guide to Java programming published by G.C.jain for Teckmedia

[10] Omid Jalilian, Abolfazl Toroghi Haghighat, AlirezeRezvanian. Evaluaion of Persian Text Based on Huffman Data Compression, Islamic Azad University, Iran

[11] Mostafa A. Bassiouni and Amar Mukherjee. Optimal Mapping for 2-bit High Speed Huffman Compression, Department of computer science, University of central Florida, Orlando

