International Journal of Latest Trends in Engineering and Technology
Special Issue SACAIM 2017, pp. 476-478
e-ISSN:2278-621X

COMPRESSED PATTERN MATCHING

Vineeth Kumar[1], Nandini D[2] & Mr.Santhosh B[3]

Abstract- Compressed Pattern Matching (CPM) is process of searching for the pattern (or set of patterns) in set of compressed text. The pattern may match or may not match the compressed text. CPM is for the most part utilized while working with extensive volume of information particularly finished the system. It additionally have applications in biotechnology, where it is utilized to discover examples of DNA groupings; interruption location over the systems, enormous information examination. There are different instance where pattern is matched directly over the uncompressed text in such solution case lot of space and time will be consumed when handling the big data. Since the extent of the compressed information is littler than that of the first information, a looking performed on the compacted information will bring about a shorter hunt time.
Keywords – Compressed Pattern Matching; Wavelet tree; Huffman compression

1. INTRODUCTION

The increase in size of data, storage and retrieval of data from big set of data is hard task. For example, in computational biology, storage of large DNA sequences requires huge disk storage. Compression is able to reduce the consuming storage space and minimize the time in further processing of the data. Compressed Pattern Matching(CPM) problem is a process of matching a pattern within the body of a compressed text, without decompressing the text. Here, pattern may be compressed.CPM is able to reduce usage of disk, compression ratio and matching time. It is useful in the networks, where large amount of data being transmitted over network which make the slower transmission of data. In this situation, CPM will help in minimize the traffic over the network and make transmission of data faster. Many algorithms have been proposed to solve the CPM.

Different scientists have proposed the efficient answers for pressure however not very many exist for design coordinating over the packed content. Considering the future pattern where information estimate is Increasing exponentially step by step, CPM has turned into an alluring undertaking. This paper introduces a Critical survey on the current methods on the packed example coordinating. The secured Techniques incorporates: Word based Huffman codes, Word Based Tagged Codes; Wavelet Tree Based Indexing. We have displayed a similar examination of the considerable number of systems said above and featured their points of interest and detriments.

The Compressed Pattern Matching issue can be expressed as: given the compacted organization of a content and an example string, report the occurrence(s) of the example in the content with negligible (or no) decompression. The principle favorable circumstances of compacted design coordinating versus the local decompress-then-look approach are: First, decreased capacity cost. Since there is no compelling reason to decompress the information or there is just negligible decompression required, the circle space and the memory cost is lessened. Second, less pursuit time. Since the measure of the packed information is littler than that of the first information, a looking performed on the compacted information will bring about a shorter inquiry time.

Most compression schemes take advantage of the fact that data contains a lot of redundant values. In addition, text space between the text will be eliminated, In image white place will be removed, audio files which has blank or silenced audio will be eliminated these are the. Compression has become crucial when combination of voice and data networks.

2. DIFFERENT TYPES COMPRESSED PATTERN MATCHING

2.1 Two important compression concepts:

- Lossy compression: Lossy compression allows loss of information is acceptable. The best example is a videoconference where the video frames will loss its clarity of images. Sometime video gets stuck or paused state because of compression standard. In the case of graphics or image files, some resolution may lose in order to generate the compressed file. This may cause in the colour, texture or resolution of image. Loss in voice and audio also accepted according to the resolution and quality.
- Lossless compression: Lossless compression, data is compressed causing no loss of data. It outputs all data back which is put in. Financial and confidential data files are examples where lossless compression is required.

[1] Dept of MCA, AIMIT, St. Aloysius College(Autonomous), Mangalore
[2] Dept of MCA, AIMIT, St. Aloysius College(Autonomous), Mangalore
[3] Dept of MCA, AIMIT, St. Aloysius College(Autonomous), Mangalore

2.2 Huffman compression based pattern matching:

A more efficient approach is to use a variable-length representation, where each character can have a different number of bits to represent it. More specifically we first analyze the frequency of each character in the text, and then we create a binary tree (called Huffman tree) giving a shorter bit representation to the most used characters, so that they can be reached faster. Notice that it must be a prefix tree (i.e., the code of every letter can't be prefix to the code of any other letter) else the decompression wouldn't work. The decoding process starts with the first bit in the input. The input bits are then used to be determined to the right or to the left in the decoding tree. When a sheet of the tree is added to a decoded character, then it places that. The bit in the incoming stream is the first character of the next character

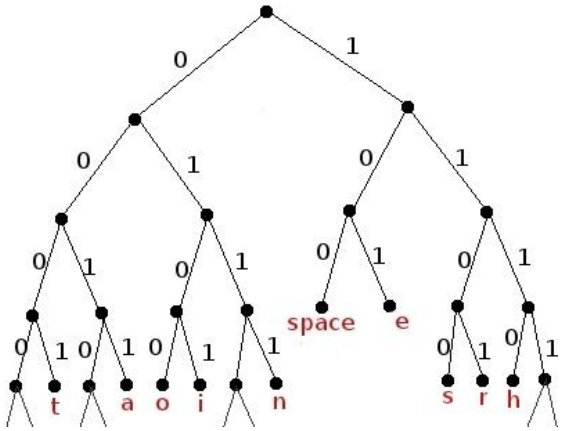

Figure 1. Huffman tree structure

2.3 Huffman compression of text:

The Huffman compression is in light year of 1952. The example we're going to use throughout this handout is encoding the particular string "happy hip hop" (don't ask me what it means, I just made it up!). Using the standard ASCII encoding, this 13-character string requires 13*8 = 104 bits total. The table below shows the relevant subset of thestandard ASCII table. To use Huffman code and to compress this sequence, it is necessary to first assemble a Huffman compression tree based on the frequency of the sequence of the character.

Character	ASCII	Bit pattern(binary)
h	104	01101000
a	97	01100001
p	112	01110000
y	121	01111001
o	111	01101111
space	32	00100000

Example table for arrangement of ASCII value and binary value for characters

The string "happy o o o" would be encoded in ASCII as 104 97 112 112 121 32 111 32 111 32 111. Though not easily readable by humans, it would be written as the following
stream of bits (each byte is boxed to show the boundaries):
01101000 01100001 01110000 01110000 01111001 00100000 01101111 00100000 01101111 00100000 01101111
To decompress or decode such a text we merely need to break the encoded stream of bits up into 8-bit bytes, and then convert each byte using the fixed ASCII encoding. The first 8 bits are 01101000, which is the pattern for number 104, and position104 in the ASCII set is assigned to lowercase 'h'. A file encoded in ASCII does not require any additional information to be decoded since the mapping from binary to characters is the same for all files and computers

2.4 Byte oriented Huffman code:

In byte arranged Huffman code, each word is encoded as entire bytes. For this situation, the Huffman tree has degree 128 (called as Tagged Huffman code) and Huffman tree has degree 256 (called as Plain Huffman code). As per, there is no hardship of the pressure proportion by utilizing byte rather than bits, while if there should arise an occurrence of byte Oriented Huffman code decompression is substantially speedier than twofold Huffman code. This uses a coding plan called as Tagged Huffman Code.

2.5 Wavelet tree based compression:

Wavelets are also used in mobile applications, denoising, edge detection, speech recognition, feature extraction, real time audio-video applications, biomedical imagining, orthogonal divisional multiplexing. So the wavelet transform popularity is

increasing due to its ability to decrease distraction in the reconstruction signal. Wavelet trees (WT) are self-indexed data structure. It reduces the cost of maintaining the text explicitly. By keeping up list of content, we can without much of a stretch track and recover a content anytime of time The wavelet change is a confined examination apparatus relative to scale. Flag investigation is performed on various scales and removals (expansions and interpretations). This occurs from a model capacity (mother wavelet) by methods for expansions and interpretations. The scaling component and interpretations by methods for a variable characterize the general state of a wavelet family. Wavelet tree change the content into adjusted paired tree. Base of the tree is doled out a bitmap. The two essential bitmap operations rank and select are utilized to execute previously mentioned operations. Rank is characterized as: rank(S, I) = k, where k is the quantity of events of image 'c' in the grouping S[1. . .i]. Characterize select(S, I) = k, where k is position of ith event of the image 'c' in the arrangement S. For instance, if S = 11001100011001, rank0(S, 11) = 5 and select1(S, 6) = 11. Standard entropy coding techniques like huffman coding permit to pack the information by doling out short codes for all the more as often as possible happening images and expansive code words for different images. By appropriate encoding, the flag requires less memory space for capacity, and takes less time amid transmission.

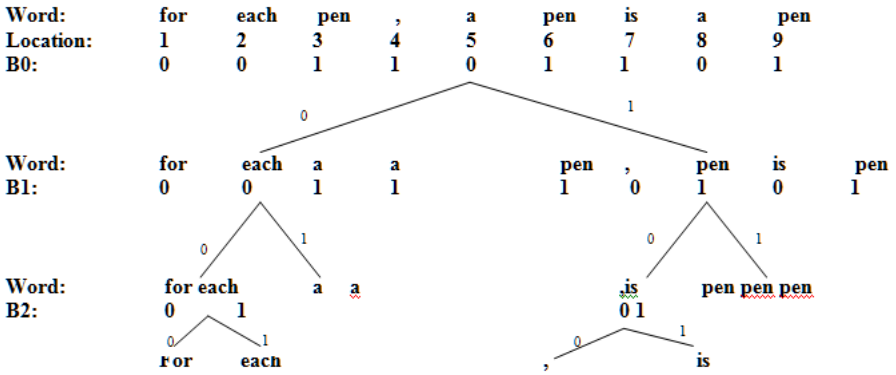

Figure 2. Wavelet Tree built from Huffman codes for words ‘‘for each pen, a pen is a pen”

The advantage of this approach is that it reduces the height of the tree and size of alphabet. Height of the wavelet tree is approximately half of the previous approach. We can retrieve the number of occurrences of word (count) using index, occurrence of a word (locate) and find word in any given position of the text (display). Assume that the word to be looked is "each". Wavelet tree of Fig. 1 demonstrates that it is encoded by 001. The operation find is utilized to seek specific in Wavelet tree. Since there is three piece code of the word, so we have to move three levels in the Wavelet tree. Since first piece is 0, so we move left offspring of root. After this, second piece is 0, so we move left offspring of root, we compute rank1 (B2, |B2|) = 1, which implies word happens one times in the leaf of right sub tree. Presently to locate the primary event of the word each, begin from leaf and move towards root. Figure select each (B2, 1) = 2 (implies first event in this hub is 2). It implies first event of word each happen at second position in hub. Presently move one name up and register select0 (B1, 2) = 2, implies second event of 0 happen at second position in hub. Presently move one name up and figure select0 (B0, 2) = 2, implies second event of 0 happen at second position in the node. Since it is root, so stop the operation. So first occurrence of word each occur at 2nd position.

3. CONCLUSION

This paper comparing efficient algorithms on Compressed Pattern Matching(CPM). The study shows that the Wavelet tree base compression pattern matching is more preferred. Compared to the Huffman Wavelet CPM technique is faster and searching the matching pattern in the compressed text. Wavelet also can be implemented check pattern matching in images also which is also efficient comparatively.

4. REFERENCES

[1] Beal, R., Adjeroh, D.A., 2013. Compressed Parameterized pattern matching. In: 2013 IEEE Data Compression Conference, March 20—22

[2] Tao Tao, Amar Mukherjee ,Compressed Pattern Matching In: 2005 Spring Term, School of Electrical Engineering and Computer Science

[3] Surya Prakash Mishra, Col.Gurmit Singh, Rajesh Prasad In: June 2016 ScienceDirect

[4] Gupta, A., Agarwal, S., 2008b. A novel approach of data compression for dynamic data. In: Proc. of IEEE Third International Conference on System of Systems Engineering, CA, USA.

[5] Prasad, R., Garg, R., 2014. Efficient multi-word parameterized matching on compressed text. In: IEEE International Conference on ICAST, 29—31 October, Ota, Nigeria