Data Compression Techniques for Big Data

[1]Ms.Poonam Bonde, [2]Mr. Sachin Barahate
[1]P.G Student, [2]Assistent Professor in I.T. Department
[1]Student of YTGOIFOE, Mumbai, India
[2]Padmabhushan Vasantdada Patil Pratishthan's College Of Engineering, Sion, Mumbai, India

***Abstract*—Due to rapidly increasing size of data which comes from different heterogeneous sources, storage, maintenance, processing & real time transfer of big data is becoming a challenging task. Many times data files contain irrelevant and redundant data which can be removed to reduce size of file manageable. Various techniques to achieve this task are called compression techniques. These techniques can be utilized to text, image, audio, video etc data. Compressed data not only increases transfer speed but also improves knowledge discovery as well as processing time.**

***Index Terms*—Compression, text compression, image compression, audio-video compression, big data compression, lossy-lossless compression**

I. INTRODUCTION

What is Big Data:

Big data means huge amount of data which maybe in structured or in unstructured format .This data is very complex and in high volume so it becomes very difficult for traditional data processing applications.

Challenges or issues due to lage size of data:

1. Storage and transport issues
2. Data Management Issues
3. Processing issues
4. Data access and information sharing
5. Relevant data searching
6. Data analytics

Compression is the process of reduction in size of data in order to save space or transmission time.

Reasons to Compress

- To Reduce File Size
- To Save storage space
- To Increase transfer speed at a given data rate
- To Allow real-time transfer at a given data rate

lossless algorithms, which can reconstruct the original message exactly from the compressed message as they don't involve loss of information, and lossy algorithms, which can only reconstruct an approximation of the original message as they involves certain amount of loss of information.

II. LITERATURE REVIEW

Compression algorithms for Text data

1. Arithmatic coding
2. Huffman Algorithm
3. LZW
4. Run length encoding

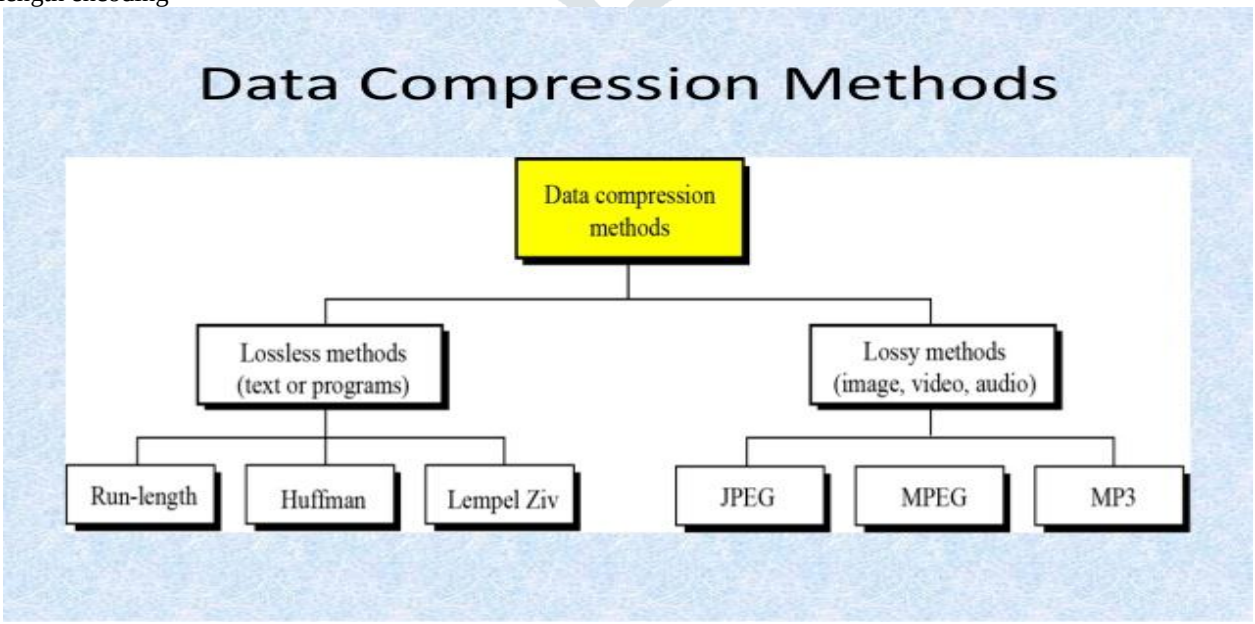

Fig. Data Compression Methods

The general purpose of data compression algorithms on text files is to convert a string of characters into a new string which contains the same information but with new length as small as possible.

Compression algorithms for image data

Methods for lossless image compression

1. Run length encoding:
2. Entropy encoding
3. Adaptive dictionary algorithms such as LZW – used in GIF and TIFF

Methods for lossy image compression

1. Chroma subsampling
2. Transform coding
3. Fractal compression

Compression algorithms for audio data

1. MPEG

Compression algorithms for video data

1. MPEG
2. H.264

Basic types of compression techniques

1. Lossy
2. Lossless[1]

Combining Compression techniques in big data

We enlisted above some issues regarding big data. Main issues are storage space & transformation cost. Using compression techniques in big data to reduce size of files can provide solution in some extent. We are focussing here on text & image compression of big data.

Text Compression

Burrows Wheeler transform

The Burrows-Wheeler Algorithm was published in the year 1994 by Michael Burrows and David Wheeler in the research report "A Block-sorting Lossless Data Compression Algorithm". This research report is based on an unpublished work by David Wheeler from the year 1983.[2]

Structure of the Burrows-Wheeler Algorithm

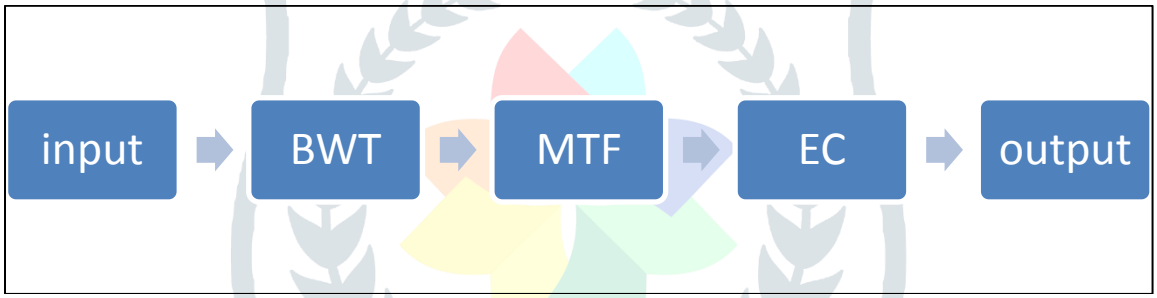

Fig. Structure of Burrows Wheeler Algorithm

The original algorithm consists of three stages.

1) Transform Coding
2) Move-To-Front Transform.
3) Zero Run Length Coding.
4) Backward Search Algorithm

Image Compression

Image compression is the process of transforming an image file in such a way that it consumes less space than the original file.
It is a compression technique that reduces the size of an image file without degrading its quality to a greater extent.

Anamorphic Stretch Transform:

An anamorphic stretch transform (AST) is a mathematical transformation in which analog or digital data is stretched and warped in a specific fashion such that after down sampling, the volume of data is reduced without loss of pertinent information.[3]

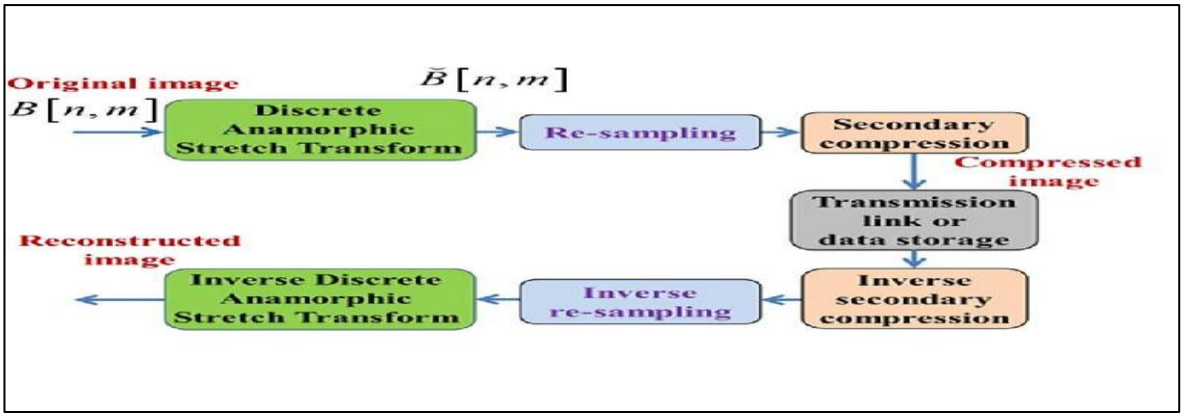

Fig. Different steps for implementation of DAST for application to image compression

The discrete implementation of this technique, dubbed Discrete Anamorphic Stretch Transform (DAST) represents a new solution to image compression [4]. The transform reshapes the image before uniform re-sampling in such a way that sharp features are naturally stretched more than coarse features. This causes sharp features to experience higher sampling density than coarse features.

III. FRAMEWORK OF OUR APPROACH

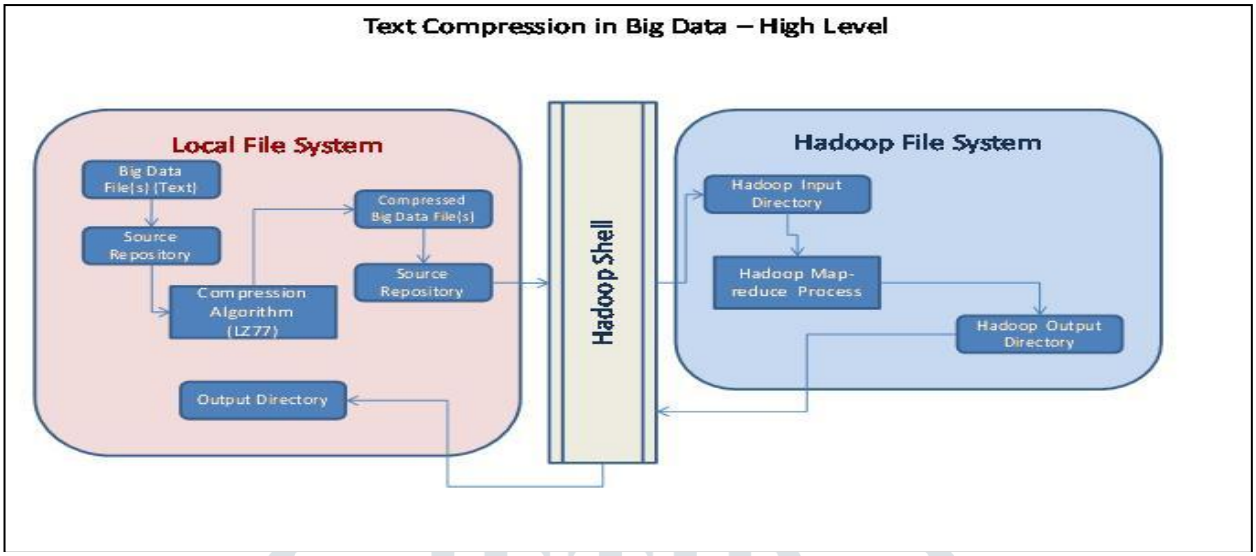

Fig . Text Compression in Big Data

Steps:

1. Get big text source file(s) (public classified information only) (input from various online social networking sites, ecommerce transaction files, supply chain transaction files, geospatial records, healthcare information etc).
2. Place the file(s) in source directory of local file system (In my case, I have used Ubuntu operating system).
3. Use a compression program (Working based on LZ77, Burrows wheeler algorithm) to compress the input file(s).
4. Using Hadoop Shell commands; transfer the compressed file to Hadoop file system.
5. Run the analytical program to analyze the files.
6. The analytics program works just fine as in case of uncompressed file (This is an observation).
7. Get the output files from the Hadoop file system to Local file system to read conveniently.

Image Compression:

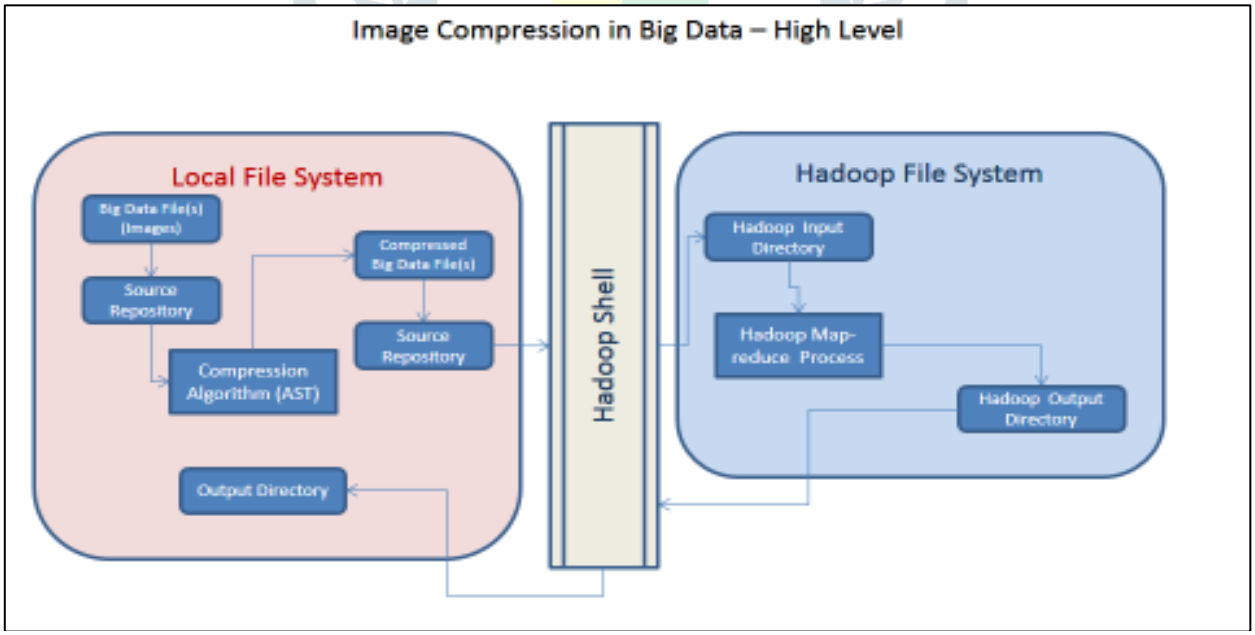

Fig. Image Compression in Big Data

Steps:

1. Get big source image file(s) (public classified information only) (input from various online social networking sites, ecommerce transaction files, supply chain transaction files, geospatial records, healthcare information etc).
2. Place the file(s) in source directory of local file system (In my case, I have used Ubuntu operating system).
3. Use a compression program which works on Anamorphic Stretch Transform to compress the input file(s).
4. Using Hadoop Shell commands; transfer the compressed file to Hadoop file system.
5. Run the analytical program to analyze the files.

Observations of using text compression techniques in Big Data using Hadoop

In my experiment, I took a large text file, using Hadoop file system, I transferred the file from local to Hadoop file system, ran a simple wordcount analytical program to check the results.

The observations are as follows

The compression technique used for following observations is modified LZ77 which is used by GZIP utility.

1. **Environment Preparation and File Transfer**

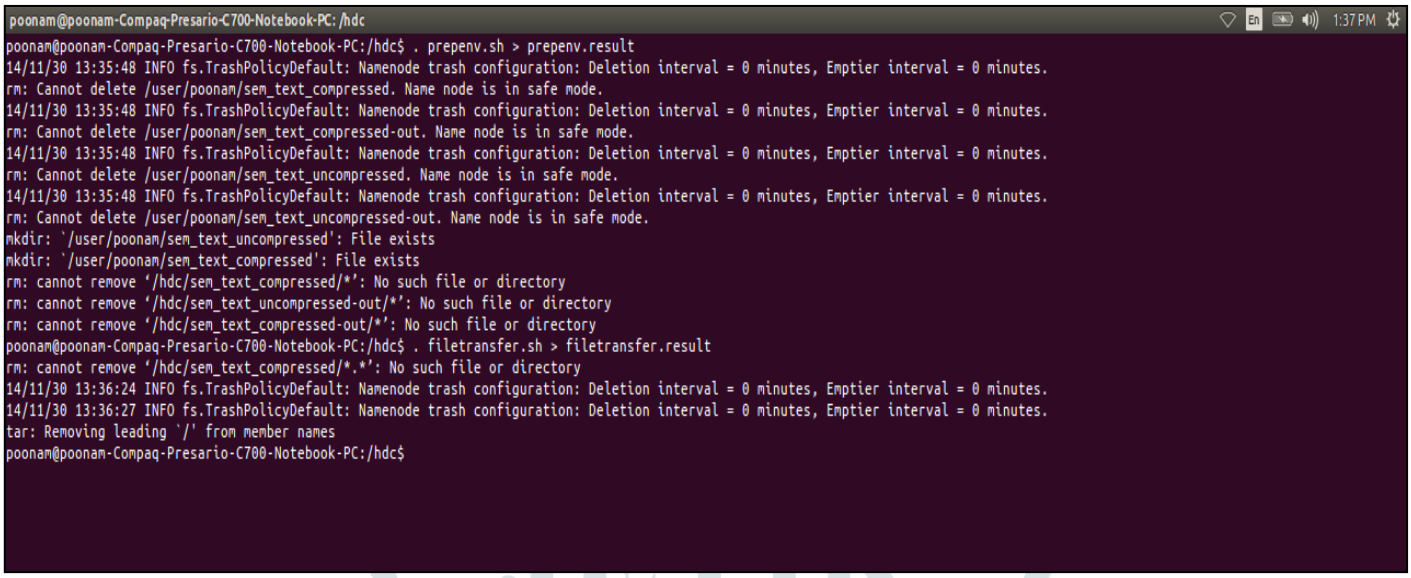

Fig. Environment Preparation & File Transfer

2. **Results of File Transfer**

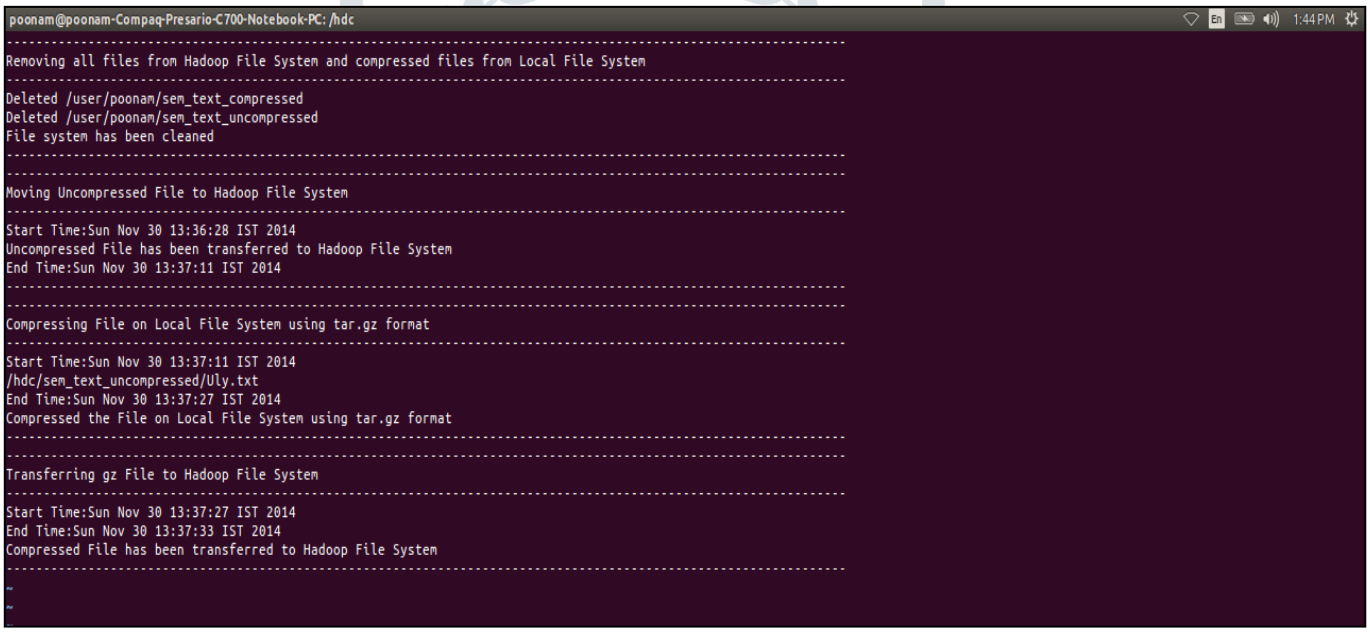

Fig. Results of File Transfer

3. **Results of running the analytics (wordcount program in this case)**

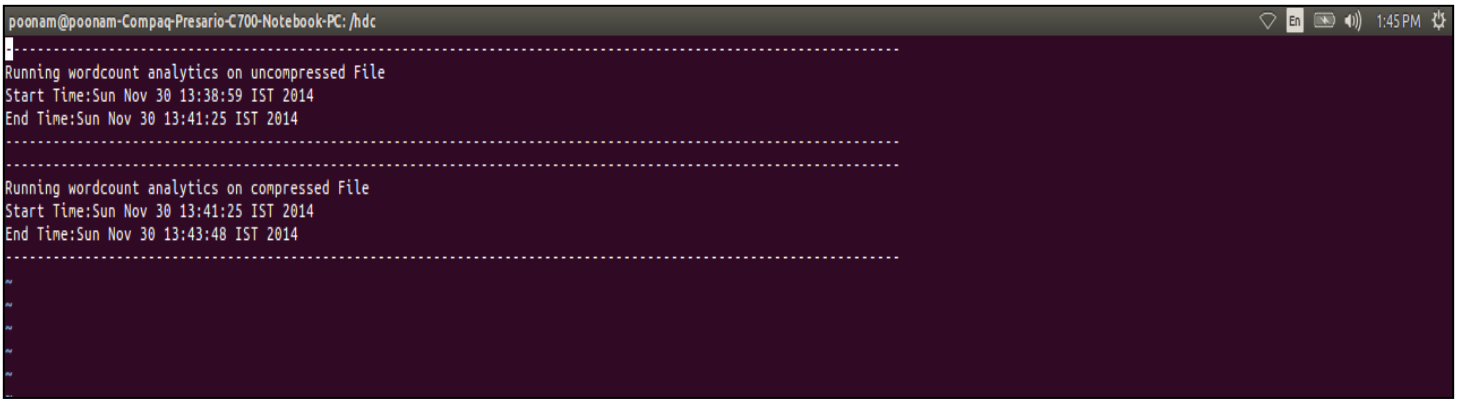

Fig. Results of running the analytics

Time

File Transfer			
File Type	Start Time	End Time	Time Taken by Process
Uncompressed	13:36:28	13:37:11	0:00:43
Compressed	13:37:27	13:37:33	0:00:06

Fig. File Transfer Time

File Compression	
Start Time	13:37:11
End Time	13:37:27
Time taken by Compression Process	0:00:16

Fig. File Compression

Analytics Processing Time			
	Start Time	End Time	Process Time
Uncompressed File	13:38:59	13:41:25	0:02:26
Compressed File	13:41:25	13:43:48	0:02:23

Fig. Processing Time

Total Time	
Uncompressed File	0:03:09
Compressed Fil	0:02:45
Time saved due to compression	0:00:24

Fig. Total Time

Size

File Size	Bytes	MBs
Before Compression	552311760	526.7255
After Compression	4743196	4.523464

Fig. Size of files

Outputs

Uncompressed File

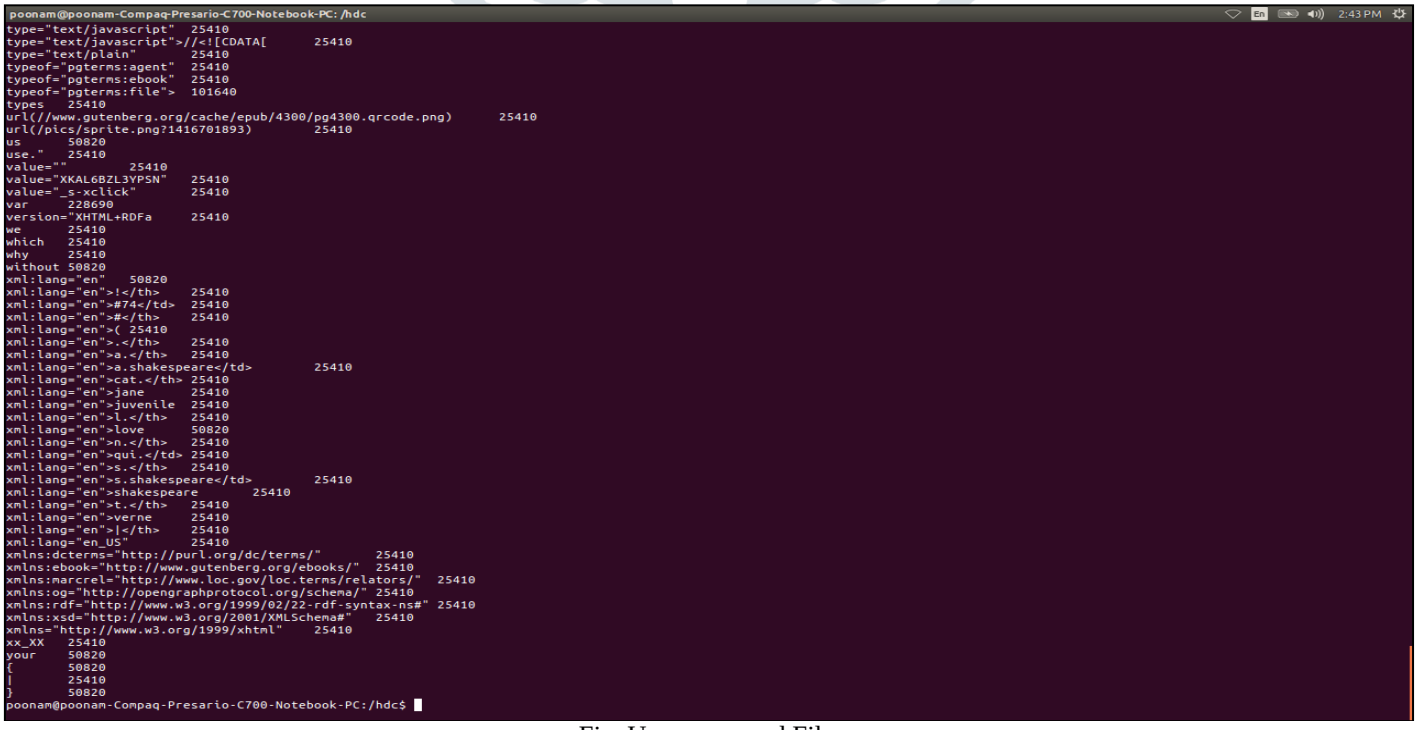

Fig. Uncompressed File

Compressed File

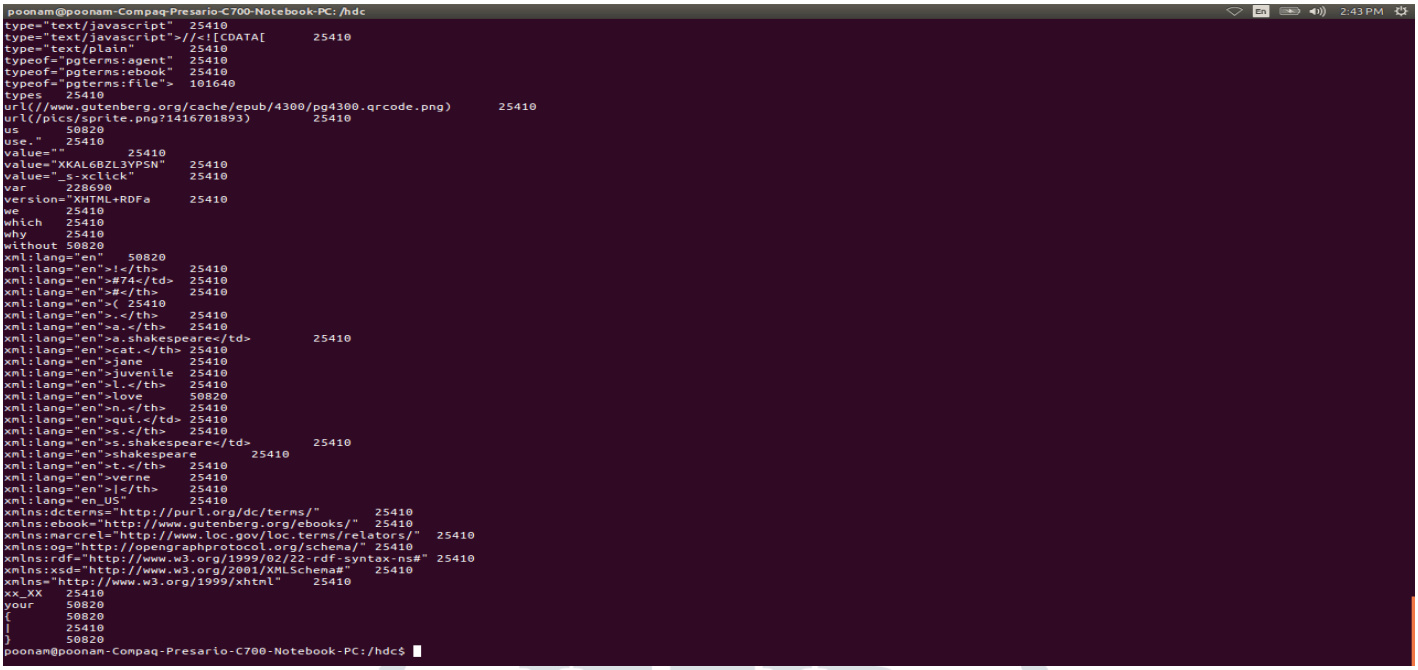

Fig. Compressed File

Results and Conclusion

1. Compressed files take less time to transfer from local system to Hadoop file system.
2. Both Compressed files and uncompressed files yield the similar results of analytics.
3. Hence use of compression techniques is feasible in Big Data by using Hadoop.
4. Following Algorithms can be used in Hadoop for text compression
    a. LZ77 (Used by Gzip in unix)
    b. Burrows Wheeler (Used by Bzip2 in unix)
    c. Along with these according to Hadoop documents, it can use Snappy.

Benefits of File Compression in Hadoop

1. Requires less storage space.
2. File transfer and ultimately the analytics are faster compared with uncompressed files.

Work Ahead

1. Using AST (Anamorphic Stretch Transform) to compress images and use it in Hadoop.

REFERENCES

[1] International Journal of Engineering Inventions e-ISSN: 2278-7461, p-ISBN: 2319-6491 Volume 2, Issue 4 (February 2013) www.ijeijournal.com
Image Compression Techniques: A Survey Athira B. Kaimal, S. Manimurugan, C.S.C .Devadass
[2] The Burrows-Wheeler Algorithm Daniel Schiller August 5, 2012
[3] http://en.wikipedia.org/wiki/Anamorphic_stretch_transform
[4] B. Jalali and M. H. Asghari, "The anamorphic stretch transform: putting the squeeze on Big Data", Optics and Photonics News Magazine, February issue cover story, Optical Society of America, Feb. 2014