The Basic Principles of Data Compression

Author: Conrad Chung, 2BrightSparks

Introduction

Internet users who download or upload files from/to the web, or use email to send or receive attachments will most likely have encountered files in compressed format. In this topic we will cover how compression works, the advantages and disadvantages of compression, as well as types of compression.

What is Compression?

Compression is the process of encoding data more efficiently to achieve a reduction in file size. One type of compression available is referred to as lossless compression. This means the compressed file will be restored exactly to its original state with no loss of data during the decompression process. This is essential to data compression as the file would be corrupted and unusable should data be lost. Another compression category which will not be covered in this article is “lossy” compression often used in multimedia files for music and images and where data is discarded.

Lossless compression algorithms use statistic modeling techniques to reduce repetitive information in a file. Some of the methods may include removal of spacing characters, representing a string of repeated characters with a single character or replacing recurring characters with smaller bit sequences.

Advantages/Disadvantages of Compression

Compression of files offer many advantages. When compressed, the quantity of bits used to store the information is reduced. Files that are smaller in size will result in shorter transmission times when they are transferred on the Internet. Compressed files also take up less storage space. File compression can zip up several small files into a single file for more convenient email transmission.

As compression is a mathematically intense process, it may be a time consuming process, especially when there is a large number of files involved. Some compression algorithms also offer varying levels of compression, with the higher levels achieving a smaller file size but taking up an even longer amount of compression time. It is a system intensive process that takes up valuable resources that can sometimes result in “Out of Memory” errors. With so many compression algorithm variants, a user downloading a compressed file may not have the necessary program to un-compress it.

Some transmission protocols may include optional compression built-in (e.g. FTP has a MODE-Z compression option), so that taking time to compress data by another process before transmission may negate some of the advantages of using such an option in the protocol (because what is eventually submitted for transmission to/by the protocol is probably now not very further-compressible at all, and may waste time while the protocol tries and fails to achieve more compression). It is distinctly possible that 'external' compression beforehand is more efficient these days, and that any compression option in the protocol should probably be deprecated. However, it is not beyond the bounds of possibility that the built-in compression actually achieves faster overall results, but possibly with larger compressed files, or vice versa. Experimentation should be employed to ascertain which applies, versus which factor is most important to the user.

History of Compression

In 1949, the Shannon-Fano coding was devised by Claude Shannon and Robert Fano to assign code words based on block probabilities. This technique was only considered fairly efficient in variable-length encodings. In 1951, David Huffman found an optimally efficient method that was better than the Shannon-Fano coding by using a frequency-sorted binary tree. Huffman coding is often used as a backend to other compression methods today.

In 1977, groundbreaking LZ77 and LZ78 algorithms were invented by Abraham Lempel and Jacob Ziv, which gained popularity rapidly. Some commonly used algorithms used today like DEFLATE, LZMA and LZX are derived from LZ77. Due to patent issues with LZ78 in 1984, UNIX developers began to adopt open source algorithms like the DEFLATE-based gzip and the Burrows-Wheeler Transform-based BZip2 formats, which managed to achieve significantly higher compression than those based on LZ78.

Types of Compression

There are several types of compression available. In the following section, we shall review the 5 types of compression offered by the backup and synchronization software, SyncBackFree, SyncBackSE and SyncBackPro.

Compression Type Available in SyncBackFree

- **DEFLATE** – Invented in 1993 by Phil Katz, this compression type is what the majority of modern compression types are based on. It utilizes LZ77 as a preprocessor combined with Huffman coding as the backend. Moderate compressed results can be achieved in a short time.

Compression Types Available in SyncBackSE (Including compression types supported by SyncBackFree)

- **DEFLATE64** – DEFLATE64 achieves better performance and compression ratio compared to DEFLATE. It is also known as Enhanced Deflate and is a proprietary trademark of PKWARE Inc.

- **Burrows-Wheeler Transform** – Known as BWT in short, this compression uses a reversible transformation technique to find repeated patterns in data and rearranging them into runs of

similar characters. With the data rearranged, BWT can then efficiently code these data, resulting in higher compression ratios.

Compression Types Available in SyncBackPro (Including all the compression types supported by SyncBackFree and SyncBackSE)

- **BZip2** – An open source variant of the Burrows-Wheeler Transform, BZip2 utilizes simple operating principles to attain a very good balance of speed and compression ratio which makes it a popular format in UNIX environments. It make use of a variety of compression techniques to produce the output. BZip2 can be very slow in certain cases, for example – handling highly random data.

- **LZMA** – The Lempel-Ziv-Markov chain-Algorithm was first released in the .7z file format with the release of 7-Zip archiver software in 1998. In most cases, LZMA is able to achieve a higher compression than BZip2, DEFLATE and other algorithms at the expense of speed and memory usage. Similar to BZip2, a chain of compression techniques are used to achieve the result.

Summary

In conclusion, data compression is very important in the computing world and it is commonly used by many applications, including the suite of SyncBack programs. In providing a brief overview on how compression works in general it is hoped this article allows users of data compression to weigh the advantages and disadvantages when working with it.