



Data Compression Benchmark and ROI Analysis

TECHNICAL WHITE PAPER

Table of Contents

Making the Case for Data Compression	3
Data Compression Primer	4
A Case for Storage ROI	6
Compression and Security	7
Putting Theory into Practice	8
Conclusion	8

Data Compression Benchmark and ROI Analysis

Making the Case for Data Compression

In today's dynamic business environment, information is the key corporate asset. While some of the information is digitally archived, much of it is accessed and exchanged internally and externally countless times using common business applications such as document editors and word processors, spreadsheets, databases, or email programs. This routine exchange of digital information has become the life-blood of many organizations.

Business communities used to be static, but in today's environment business communities are dynamic and often even virtual. In these dynamic environments, information is likely to travel through one or more networks. The increased reliance on the steady transfer of information is being driven by distributed organizations, extended supplier-manufacturer networks, and customer relationship management applications.

Internally, constant data transfer is required to keep all of the critical applications operating efficiently. Sending daily reports from the zSeries to regional offices, updating corporate inventory systems from local offices, and ubiquitous email communications are all data-driven activities happening daily or even hourly within enterprise companies today.

Most businesses are accustomed to the convenience of streamlined communications, but many are unaware of what takes place below the surface. Few users realize how their actions impact the budget and ROI of the IT organization.

Hidden Costs

All of the information entered into a document, regardless of the application or format, becomes digital data. Once a file is created, its data must be saved within the storage facility of the underlying computer system. If that file is exchanged with co-workers via email, for example, it will most likely be replicated on a number of other systems, depending on how many people received a copy of the file. It will also end up in the mail server storage facility, and ultimately, in archival storage.

Digital storage and transfer of information assets come at a price, both in terms of transmission bandwidth and storage costs. Most organizations must buy or lease sufficient storage capacity to house all of the data generated and used by their extended network. Many technology options exist today for this task, yet all of these options result in recurring costs that can increase with the amount of information exchanged.

In addition to impacting costs, any time data is transmitted, the risk of corruption is increased. Compressing data can help reduce this risk, as well as save space. The smaller the file, the less likely that data integrity will be compromised in transmission. A good compression product can also check data after transmission to ensure that the data received matches exactly with the data sent.

Data Compression Primer

The purpose of data compression is simply to make files smaller. During compression, reduction in a file's size is achieved by eliminating redundant patterns and by encoding the contents of the file using symbols that require less storage space than was originally required. After a file is compressed, its content is changed to an encoded form, and the file cannot be used until it is decompressed. The decompression process is the inverse of compression. It restores a file to its original state. As a simple example of the data compression process, consider this sentence:

she sells sea shells by the sea shore

This sentence consists of 37 characters, including spaces. The spaces are important and cannot be simply thrown away since removal would change the meaning of the original message. The science of compression recognizes the repeating patterns in this sentence. The combination 'se' appears three times, 'sh' three times, and 'lls' twice. In fact, the 'se' pairs all have a space in front of them, and can be represented by ' se'. These patterns define the redundancy of the message.

she sells sea shells by the sea shore

Each of these patterns can be encoded by replacing them with a single character. For example:

= " se"
 |\$ = "sh"
 % = "lls"

Note that the first replacement string includes a space at the beginning. The new form of this sentence using these symbols looks like this:

\$e#%#a \$e% by the#a \$ore

The new representation is 24 characters long. This is a saving of 13 characters, or 36 percent. Applying the decompression process to the compressed string would result in the replacement symbols being converted back into their original form, restoring the original message.

How does it work?

Compression is accomplished using algorithms developed to achieve the best compression results for a given type of file. Algorithms are computer programs written to complete the steps needed to compress a file. Most compression algorithms operate in a more complex manner than the example above, but all operate by the same principle of replacing repeating patterns with efficient encoding methods.

Different file types will typically compress to different sizes. The amount of compression that may be gained for any file, regardless of type, depends on how much redundant information it contains. Files with a high level of redundancy will compress more than files with low redundancy.

There are two main types of compression: lossy and lossless

Lossy compression assumes that it is okay to discard some of the original data in order to achieve more efficient compression. This means that after the file is decompressed, it is not an exact copy of the original since some of the data was "lost" by the compression process. This assumption is valid for some types of data such as images. For example, when a file representing an image is compressed with lossy compression, the process may actually discard some of the image data to achieve a greater compression rate. Typically the human eye cannot detect the differences between the image generated from the original file and the image generated from the decompressed file. Where a loss of data is acceptable, compression rates well over 90 percent are common.

Lossless compression takes the opposite approach. In lossless compression, it is unacceptable to discard any data and the decompressed form of a file must exactly match the original file content. Lossless compression is used when it is necessary to faithfully reproduce the contents of a file through decompression. Files containing words or numbers, such as financial data, and files that are intended for further computer processing may require lossless compression. In these situations, it would not be acceptable for any of the content to be discarded or altered by the compression process.

Another important component of compression technology is the trade-off that exists between time and file size reduction. The more time available to remove the redundancy from a file, the smaller the resulting compressed file can be. However, if less time is provided, less compression

can be achieved. This is where controlled benchmarks can differ dramatically from real customer examples. An algorithm that may claim to provide superior compression may not be practical to use if it takes far too long to complete the compression process.

Compression Analysis on Common File Types and Datasets

As a practical example, the tables that follow illustrate the potential space savings to be gained using data compression. The files used for this analysis represent a random sampling of the types

and sizes of files that may be found on an average desktop, midrange, or mainframe computer.

These files were all compressed using PKZIP on various computing platforms, configured for maximum compression. PKZIP uses a lossless compression algorithm called “Deflate” that was developed by PKWARE. Deflate is a general purpose algorithm suitable for compressing most files by more than 50 percent. The benefits of data compression are not limited to any specific operating system.

Although the data used in this analysis is based primarily on common Windows desktop files and midrange/mainframe data sets, compression is also available on other platforms including Windows Server and UNIX/Linux servers. Similar compression results as those shown below can be achieved on these platforms as well.

To understand data compression, it is important to know the original size of a file and the resulting size of the file after it is compressed. With this information, the ratio of the original file size to the compressed file size can be calculated. This ratio represents the space savings from compression. The formula used to calculate the Space Savings value in the tables below is:

$$\text{Space Savings} = 100 - ((\text{Compressed Size} * 100) / \text{Original Size})$$

Results of Compression Analysis

The amount a file can be compressed depends on how much redundant information can be removed by the compression process.

It can also be seen that regardless of the type of file, compression can significantly reduce the space required for storage or transmission. This should not imply that every compressed file will result in space savings of well over 50 percent. Some file types are better candidates for compression than others. Specifically, rich media file formats generally provide little additional space savings by applying an external compression process. This is because the internal storage format for these files already includes compression. A file that has already been compressed has had most of the redundancy removed and it is unlikely that additional compression will result in a significantly smaller file.

Space savings are listed as a percentage. The benefits of data compression are not limited by the size of a file. Compression creates cost savings by storing large files (such as a 20 GB file) in a smaller “footprint” where sometimes the resulting file is reduced to a size that is 70 percent smaller, thereby reducing the overall amount of disc space being used.

zSeries and iSeries Compressed Data

File	Original Size (bytes)	Compressed Size (bytes)	Space Savings
Inventory DB	14,433,440	1,218,919	92%
SYS1.MACLIB	132,525,040	20,078,537	85%
C source library	122,964,261	18,929,705	85%
C Listings	204,188,840	21,051,729	90%
Load Library	623,362,048	138,843,804	78%

.XML - Extensible Markup Language Format

File	Original Size (bytes)	Compressed Size (bytes)	Space Savings
Index.xml	28,362	9,248	67%
Librarian.xml	29,362	8,568	70%
Lockergnome.xml	18,222	4,215	76%
Pr.xml	9,153	2,844	68%
Rss.xml	21,676	8,009	63%

.PDF - Adobe Portable Document Format

File	Original Size (bytes)	Compressed Size (bytes)	Space Savings
CryptoC6.0_Ref.pdf	3,191,560	1,555,154	51%
GDP.pdf	74,810	14,972	79%
Palm OSReference.pdf	10,353,758	14,972	52%
Soa.pdf	1,030,232	422,230	59%
UIGuidelines.pdf	1,218,247	642,580	47%

.DOC - Microsoft Word Document

File	Original Size (bytes)	Compressed Size (bytes)	Space Savings
AppendixH.doc	642,560	94,097	85%
Certificate_Testing.doc	1,503,744	198,426	86%
Climan.doc	1,668,608	271,457	83%
D1.0.doc	153,088	23,307	84%
Milan.doc	131,584	33,186	74%

The Bandwidth Conundrum: Bigger Files Require Bigger Pipes

With the volume of network traffic growing exponentially, heavy traffic can affect both the expediency and cost of sending critical data files. There are a number of network connection types in use today. The time and cost to move data varies by the type of connection. But bandwidth is not free.

One way to look at the cost of bandwidth is to compare the throughput and costs of compressed data with the cost of non-compressed data. In the chart below, we examine how much bandwidth an organization would save if they compressed the data before sending it.

Amount of Data That Could be Compressed with PKZIP Per Year

Level of Compression	100 Gigabyte	500 Gigabyte	750 Gigabyte	1 Terabyte	2 Terabyte	3 Terabyte
75%	\$72,000	\$96,000	\$120,000	\$144,000	\$168,000	\$192,000
80%	\$90,000	\$120,000	\$150,000	\$180,000	\$210,000	\$240,000
85%	\$120,000	\$160,000	\$200,000	\$240,000	\$280,000	\$320,000
90%	\$180,000	\$240,000	\$300,000	\$360,000	\$420,000	\$480,000
95%	\$360,000	\$480,000	\$600,000	\$720,000	\$840,000	\$960,000
99%	\$1,800,000	\$2,400,000	\$3,000,000	\$3,600,000	\$4,200,000	\$4,800,000

The chart above is based on industry-standard costs for bandwidth.

A Case for Storage ROI

Specifically, a cost of \$18,000 per year for a T1 leased line was used to demonstrate the comparison of the costs to send a compressed file and an uncompressed file in the same period of time. Your costs may differ.

For example, an organization sending 1TB of data per year, achieving an 85 percent compression rate, would save \$240,000 in bandwidth costs (compared to the cost of increased bandwidth).

Source: IDC Document #34357, Worldwide Disk Storage Systems 2005-2009 Forecast Update: Midyear Update, Nov 2005, Dave Reinsel, Natalya Yezhkova, Brad Nisbet

**TABLE 4
Worldwide Disk Storage Systems Terabytes by Storage Location, 2004-2009**

	2004	2005	2006	2007	2008	2009	2004-2009 CAGR (%)
External	784,310	1,240,800	1,999,067	3,187,524	5,070,032	8,027,294	59.2
Internal	499,065	705,518	969,392	1,327,234	1,832,092	2,530,584	38.4
Total	1,283,375	1,946,319	2,968,459	4,514,758	6,902,124	10,557,878	52.4

Data storage has become critical infrastructure to support information technology. The information generated and used on a day-to-day basis needs to be stored for future access and the requirements for how long this information must be available are increasing. The need to archive information comes from many sources, including organizations that need to recover customer or other business information on demand, as well as regulations that require corporate information such as email to be recoverable over extended periods of time. In general, as more and more information moves into the digital world, the need for storage continually increases. IDC's projected growth in storage through 2009 is summarized in the graph above.

A limiting factor on storage is capacity. All storage facilities are limited by the amount of information they can physically hold. As capacity is reached, information can be either deleted from the system, selectively moved to another location, or additional capacity must be added. With the increasing need to retain more data for longer periods, simply deleting data is generally not a viable option for most organizations. In practice, a combination of alternate locations and additional capacity is used. Storage, like bandwidth, also has a cost. Despite the decreased costs for storage in recent years,

the cost of purchasing and maintaining an enterprise storage facility is still a significant expense. Many storage devices today cost upwards of several hundred thousand dollars and require ongoing operational and management costs. It makes sense to store compressed data in order to get the most out of each storage dollar. This is why compression has become an essential tool in managing the inherent capacity limitations and costs of storage devices. With compression, more data can be made to fit within each megabyte, making more effective use of each available megabyte of storage.

A Sample Organization's Savings

The following simple ROI Storage Calculator is able to quantify

how compression can translate into storage savings for an organization. This calculator will help you estimate the potential savings per year in average storage costs achieved through the use of compression. This model will focus on e-mail attachment storage as one of the most ubiquitous areas where storage is needed within the enterprise.

To use this calculator, determine the total number of email users within your organization and insert that number as appropriate.

To Calculate Email Attachment Storage Savings Per Year:

1. (Number of Users) x (75 MB x \$0.53) = (Storage cost per year)
2. (Number of Users) x (37.5 MB x \$0.53) = (Compressed storage cost per year)
3. Subtract the value from line 2 from the value from line 1: (Storage cost per year) – (Compressed storage cost per year) = Savings per year

The example above uses industry-standard costs and estimates for email applications. The figure 75 represents an estimate of the annual storage space used by an average employee, and \$0.53 is the average per MB cost of mail storage. Our example assumes a 50 percent reduction in file size after compression. These figures were used to illustrate how compression can save storage costs. Your costs may vary.

In addition to the significant savings in storage and bandwidth that can be achieved with data compression, it also provides a level of security to critical data. In today's world, the increased need for security has been globally recognized. This need arises from many factors. Govern-

ment regulations such as HIPAA, Sarbanes Oxley, and GLBA are driving organizations to implement security solutions; and, in the data center, security is necessary to protect datasets residing on backup tapes and other media that are sent to off-site storage facilities. The increased need to secure sensitive information in an increasingly networked world requires new methods of achieving data security not previously considered by most enterprise environments.

Compression and Security

Securing digital information typically takes the form of data encryption. The

purpose of data encryption is to scramble the contents of a message or file so it cannot be understood by anyone unless they have a specific key or access code. The key or access code is required to "unlock" the data, allowing authorized recipients to decrypt the message or file back into its original form. This is similar in concept to compression, which as we've seen, encodes data for the purpose of reducing its size. Encryption encodes data for the purpose of making it unreadable.

Encryption is applied to data using encryption algorithms. A number of encryption algorithms are available to choose from. Examples of common algorithms include 3DES and AES. AES stands for Advanced Encryption Standard and is the algorithm selected by the U.S. Government as its new encryption standard.

Combining data compression with encryption offers several key benefits to file security. If a file is compressed before it is encrypted, the encryption process can generally complete more quickly than if the same file is encrypted without compression. This is because the total amount of data to encrypt is less after compression.

The resulting encrypted file will be smaller if it is compressed than if it is only encrypted. In many cases, applying encryption without compression will result in increasing the size of the file.

Compressing a file before encrypting further scrambles the data before the file is encrypted. This can make it more difficult for a hacker to crack since the additional compression encoding increases the work factor required to decrypt a file. The work factor is the amount of time and effort a hacker is willing to spend trying to gain unauthorized access to a file. To help put the work factor into perspective, the website of the National Institute of

Standards and Technology (NIST; www.nist.gov) lists the time to crack an AES encrypted file using a 256-bit key as 149 trillion years.

To illustrate the benefits of combining compression with encryption, a set of files from the previous compression example will be used to show the affect of applying just encryption to the files using the standard S/MIME encryption commonly found in many commercial email products. This will be followed by combining both compression and encryption to the same file set. The purpose of this example is to show the benefit compressing a file before encrypting has on the resulting size of the compressed and encrypted file.

The first example will use 3DES 168-bit encryption as applied using S/MIME. In the second example, PKZIP will be configured to use maximum compression before encrypting with 3DES 168-bit. Both example file sets were mailed as file attachments using Outlook 2002.

The data above shows that when using encryption alone on each file, the resulting sizes of the encrypted files mailed are actually significantly bigger than their original size. S/MIME encryption typically adds overhead. If the same files are compressed before being encrypted, each file is significantly smaller than when encryption is used alone. Each file is less than half the size of the original.

.DOC - Microsoft Word Document with Encryption Only

File	Original Size (bytes)	Compressed Size (bytes)	Difference (%)
AppendixH.doc	642,560	893,440	39%
Certificate Testing.doc	1,503,744	2,357,760	56%
Climan.doc	1,668,608	2,337,280	40%
D1.0.doc	153,088	211,968	38%
Milan.doc	131,584	193,536	47%

The results show the space savings gained when using a common file type on PCs. Similar space savings are achieved when using compression on other platforms, including

zSeries, iSeries, Windows Server, and UNIX/Linux servers.

The key to realizing the ROI of data compression is through the integration of compression technology into business processes and software applications. Compression technology is available in many forms. A choice can be made from a range of efficient hardware or software based compression options. Compression algorithms exist today that can be integrated into hardware or software and in some cases both. There are algorithms for compressing specific data types and there are algorithms that compress almost any data type. Some algorithms require extensive changes to integrate with existing processes and applications, while others integrate easily.

Which option is best depends on a number of requirements that vary by organization. These requirements include factors such as cross-platform support, hardware needs, software needs, and data needs and cost, among others.

A well-known and trusted data compression product available today is PKZIP® developed by PKWARE®, INC. Based on the standard .ZIP compressed file format, PKZIP provides a cross-platform solution that is designed to easily integrate into existing processes

.DOC - Microsoft Word Document with Both Compression and Encryption

File	Original Size (bytes)	Compressed Size (bytes)	Space Savings
AppendixH.doc	642,560	120,832	81%
Certificate Testing.doc	1,503,744	225,792	84%
Climan.doc	1,668,608	299,520	82%
D1.0.doc	153,088	49,152	67%
Milan.doc	131,584	59,392	54%

and applications to quickly gain the benefits of compression with less costly overhead. PKZIP provides industry-leading compression across a wide range of file types and consistently outperforms other compression solutions.

Putting Theory into Practice

A key benefit of PKZIP is the ease with which

it integrates into various computing platforms. It can be quickly automated or batched using the PKZIP command line utility to compress files into the .ZIP format for storage or transfer. Data on a zSeries system can be batched and then sent via FTP to Windows, UNIX/ Linux, or iSeries® systems. The .ZIP file format offers a significant advantage over other platform-specific compression options. As a portable format, it can be easily moved from system to system to efficiently transfer data.

Conclusion

Data compression offers a technology solution for increasing the efficiencies and decreasing

the costs of storing and transferring critical business information. With average compression rates of 50 percent per file, it makes sense to integrate compression technology into business processes and applications. PKZIP compression solutions offer trusted technologies to help take full advantage of the benefits data compression can bring to the enterprise.

References

The Data Compression Book 2nd Edition; by Nelson, Mark and Gailly, Jean-Loup; M&T Books 1996 ISBN 1-55851-434-1 NIST AES Fact Sheet; <http://csrc.nist.gov/CryptoToolkit/aes/aesfact.html>

United States
648 N. Plankinton Ave.
Suite 220
Milwaukee, WI 53203
1.888.4.PKWARE

APAC
Cerulean Tower 15F 26-1
Sakuragaoka-cho, Shibuya-ku
Tokyo 150-8512 Japan
+81.3.5456.5599

UK/EMEA
Crown House
72 Hammersmith Road
London W14 8TH
United Kingdom
ph: +44 (0) 207 470 2420