

Realtime
publishers

"Leading the Conversation"

The Shortcut Guide[™] To



Managing Disk Fragmentation

sponsored by



Mike Danseglio

Introduction to Realtime Publishers

by Don Jones, Series Editor

For several years, now, Realtime has produced dozens and dozens of high-quality books that just happen to be delivered in electronic format—at no cost to you, the reader. We’ve made this unique publishing model work through the generous support and cooperation of our sponsors, who agree to bear each book’s production expenses for the benefit of our readers.

Although we’ve always offered our publications to you for free, don’t think for a moment that quality is anything less than our top priority. My job is to make sure that our books are as good as—and in most cases better than—any printed book that would cost you \$40 or more. Our electronic publishing model offers several advantages over printed books: You receive chapters literally as fast as our authors produce them (hence the “realtime” aspect of our model), and we can update chapters to reflect the latest changes in technology.

I want to point out that our books are by no means paid advertisements or white papers. We’re an independent publishing company, and an important aspect of my job is to make sure that our authors are free to voice their expertise and opinions without reservation or restriction. We maintain complete editorial control of our publications, and I’m proud that we’ve produced so many quality books over the past years.

I want to extend an invitation to visit us at <http://nexus.realtimepublishers.com>, especially if you’ve received this publication from a friend or colleague. We have a wide variety of additional books on a range of topics, and you’re sure to find something that’s of interest to you—and it won’t cost you a thing. We hope you’ll continue to come to Realtime for your educational needs far into the future.

Until then, enjoy.

Don Jones

Introduction to Realtime Publishers.....	i
Chapter 1: Introduction to Disk Architecture	1
Introduction to Disk Architecture	2
Hard Disks and Disk Architectures.....	4
Disk Interfaces	5
IDE.....	5
SCSI	6
SATA	9
Disk Interface Wrap Up.....	9
Fault-Tolerant Disk Systems.....	10
RAID 0.....	10
RAID 1	11
RAID 5.....	11
RAID Wrap Up.....	11
File Systems	12
FAT	12
FAT12.....	13
FAT16.....	13
FAT32.....	13
NTFS.....	13
Other PC File Systems	14
HPFS	14
ext3.....	14
File System Wrap Up.....	15
How Disks Are Used	15
Fragmentation	16
Disk Bandwidth	17
Summary	17
Chapter 2: Issues with Disk Fragmentation	18
Negative Impacts of Disk Fragmentation	19
Performance	19
Common Fragmentation Scenarios.....	20
Newly Set Up Computer.....	20

Computer that Has Been Running a Long Time.....	21
File Server.....	22
Computer with a Full Hard Disk.....	24
Data Backup and Restore.....	24
Data Backup.....	25
Disk to Tape.....	26
Disk to Disk.....	26
Disk to Disk to Tape.....	27
Disk to Optical.....	28
Data Restore.....	28
Stability.....	29
Boot Failure.....	29
Program and Process Failure.....	30
Media Recording Failure.....	30
Premature Hardware Failure.....	30
Memory-Based System Instability.....	31
Summary.....	31
Chapter 3: Solving Disk Fragmentation Issues.....	32
Performance.....	33
Backup and Restore.....	34
Stability.....	35
Addressing the Disk Fragmentation Problem.....	36
Evaluating a Defragmentation Solution.....	36
Cost.....	37
Defragmentation Engine Operation.....	37
Deployment.....	39
Operational Autonomy.....	40
User Experience.....	41
Reporting.....	44
Defragmentation Approaches.....	45
Automatic Defragmentation.....	45
Manual Defragmentation.....	46
How to Make Your Decision.....	46

Preselection	47
Test.....	47
Purchase	48
Deployment.....	49
Summary	50
Chapter 4: The Business Need for Defragmentation	51
Understanding the Investment	52
Performance	52
User-Perceived Performance Issues.....	53
Less-Perceived Performance Issues.....	54
Example of Fragmentation Performance Impact	54
Data Integrity	55
Stability.....	56
Justifying the Investment	57
Performance	57
Improving Employee Productivity.....	57
Increasing System Longevity.....	58
Decreasing Deployment Time	59
Data Integrity	59
Stability.....	60
Cost-Benefit Analysis Summary.....	60
How to Make Your Decision	61
Preselection.....	61
Test.....	62
Purchase	63
Deployment.....	63
Summary	64
Glossary	65

Copyright Statement

© 2008 Realtime Publishers, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtime Publishers, Inc. (the "Materials") and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtime Publishers, Inc or its web site sponsors. In no event shall Realtime Publishers, Inc. or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtime Publishers and the Realtime Publishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtime Publishers, please contact us via e-mail at info@realtimepublishers.com.

Chapter 1: Introduction to Disk Architecture

Computers were initially designed to store and process data. Little has changed in this regard since the invention of the modern computer in the mid-20th century. However, the scale has increased tremendously. Computers process an immense amount of data, and that data must be stored somewhere. On modern computers, that storage is usually a hard disk.

Storing data on a disk has become less expensive and more convenient in modern times. Hard disk prices are, at the time of this writing, incredibly inexpensive. 750GB of hard disk storage, which just 5 years ago required a large disk array and cost tens of thousands of dollars to plan and implement, costs less than \$400 for a single off-the-shelf drive unit.

But with that increase in storage capacity and decrease in price comes a problem of management. Unfortunately, most modern administrators are complacent about their hard disks. Little care is taken to ensure that the disks continue to perform at their best based on numerous recent Gartner Group surveys. But with very little work, these disks can be maintained in optimal condition and provide exceptional performance for years.

This guide will explore the storage of data on hard disks. This chapter will examine how modern computers and operating systems (OSs) implement disk storage and access. Later chapters will explore one pervasive problem with such storage—fragmentation. This guide will also examine how fragmentation affects a computer system and what approaches are effective in reducing the effects. This guide is comprised of four chapters:

- Chapter 1: Introduction to Disk Architecture—Explains disk structure and storage techniques at a basic level. This information is essential to understanding the problems of modern data storage, including disk fragmentation.
- Chapter 2: Issues with Disk Fragmentation—Describes why disk fragmentation is a problem. Although performance is usually cited as the only fragmentation-related problem, there are a number of symptoms that can result from a fragmented disk.
- Chapter 3: Solving Disk Fragmentation Issues—Explores the most effective methods to both prevent and remediate disk fragmentation issues. Each remedy is analyzed to help you decide which is best for your environment.
- Chapter 4: The Business Case for Defragmentation—Analyzes the decision-making parameters around fragmentation. Cost/benefit analysis is performed from several viewpoints to help you make a decision about how to solve this problem.

Introduction to Disk Architecture

To understand disk performance, it is necessary to take a brief look at disk architecture, which is usually a misunderstood topic as it can be technically complex and has a number of variables that change as technology develops. You should have a basic understanding of disk operations to help you make the right choices in your disk management strategy. Thus, this guide will introduce you to the most basic and universal concepts that will help you understand the business problem and possible solutions. This guide is not intended to be a compendious reference to disk architecture.

The disk is connected to the computer through several layers of connectivity that make up the data pathway. The disk itself has its own controller and storage architecture. For the computer to understand the disk storage, a common interface must be defined. This interface is the disk interface. For the OS to communicate with the disk interface (and hence the disk architecture), some type of intermediate system must be in place. This is the file system. Each of these elements is discussed in this section.

Figure 1.1 illustrates most of the components in the data pathway between an application, such as Microsoft Word, and the actual disk subsystem that stores and retrieves its data. The graphic is slightly simplified to omit some of the less-important parts of the data pathway. The blue arrows indicate transitions from one transmission media to another and are common disk throughput *bottlenecks* (described later).

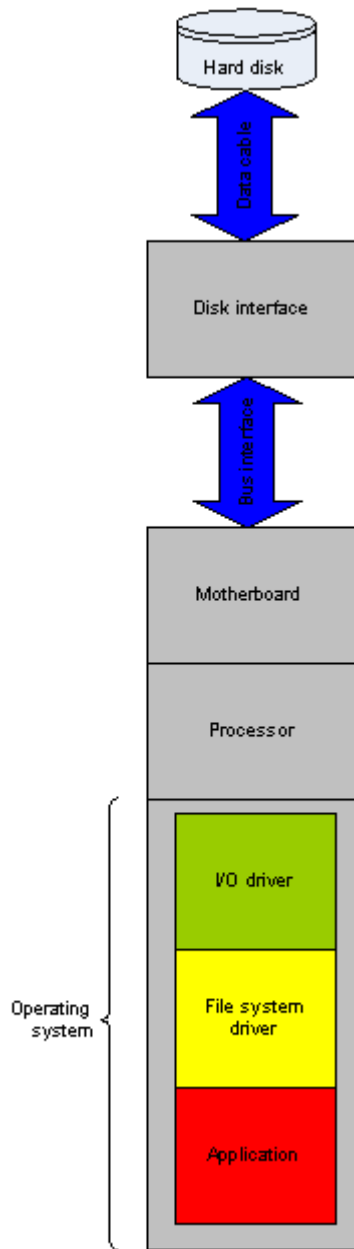


Figure 1.1: The data pathway between an application and a hard disk.

Hard Disks and Disk Architectures

Hard disks have evolved greatly over the past few years and are very different from the earliest examples in 1955. At their core, they are simply persistent data storage devices. Data can be stored on them and retrieved later. For the purposes of this guide, you need to know only a few things about the hard disk itself.

A hard disk has one or more platters, which are circular surfaces that store data. The data is written to and read from the surfaces by a read/write head, or simply head. The platters spin very quickly while the head moves over them closely, reading and writing the data as the platters spin. The head actually applies a magnetic field as it moves across the disk, which is a smooth magnetic surface. The data is stored as 0s and 1s corresponding to whether the point on the disk is magnetized or not (see Figure 1.2).



Figure 1.2: A hard disk spinning while the head accesses the data (Source: <http://www.flickr.com/photos/alphasix/179676660>).

The speed of the spinning platters, the head, and the interface all contribute to the speed of the disk. For that reason, disks are often listed with those speeds as major selling points. For example, disks with platters that spin at 10,000RPM are priced higher than disks with the same storage capacity that spin only at 7200RPM or 5400RPM. The higher RPM usually corresponds to faster disk throughput. (Unfortunately, it also often corresponds to more heat generation and power consumption.)

There is one common misnomer that should be made clear to avoid future confusion. Disks store data in units called sectors. One sector is the smallest part of a disk that can be discretely addressed. When you occasionally see an error such as “Sector not found,” the error is referring to that portion of the physical disk.

Clusters are logical groups of one or more sectors. Clusters are normally defined by the file system, which is implemented by the OS. They help optimize the use of disks by abstracting applications from the physical disk geometry. The abstraction allows the application developer to concentrate on simpler read and write applications without having to know detailed information about disk subsystems, and it allows the OS more complete and efficient control over the disks.

Disk Interfaces

The connection between the hard disk and the motherboard is called the disk interface. The data coming from the computer’s processor to the disk must be converted to a transmission method that the disk can understand, and the same transformation must be made when data goes from the disk to the motherboard.

Over the years, disk interfaces have changed radically—in fact, there has been substantially more change in disk interfaces than in the disks themselves. Each has its benefits and drawbacks. There are currently three popular interfaces in widespread use: Integrated Drive Electronics (IDE), Small Computer System Interface (SCSI), and Serial Advanced Technology Attachment (SATA).

IDE

IDE (also frequently called ATA) is an older and well-established disk interface. Most computers support IDE, and a variety of low-cost drives are available with this interface. IDE has developed over the past several years and now supports increased data throughput speeds.

The main failings with IDE drives are their limited throughput and cumbersome cabling. The throughput of IDE has been increased over the years with backward-compatible hardware upgrades. Today’s IDE drives can sustain 133MBps, and that speed is shared among all devices attached to the same IDE connector. Although this speed is faster than the original IDE devices, it is far slower than the now-common 3GBps available with SATA.

Cumbersome cabling has always plagued IDE. It is most often implemented as a flat ribbon cable as shown in Figure 1.3.

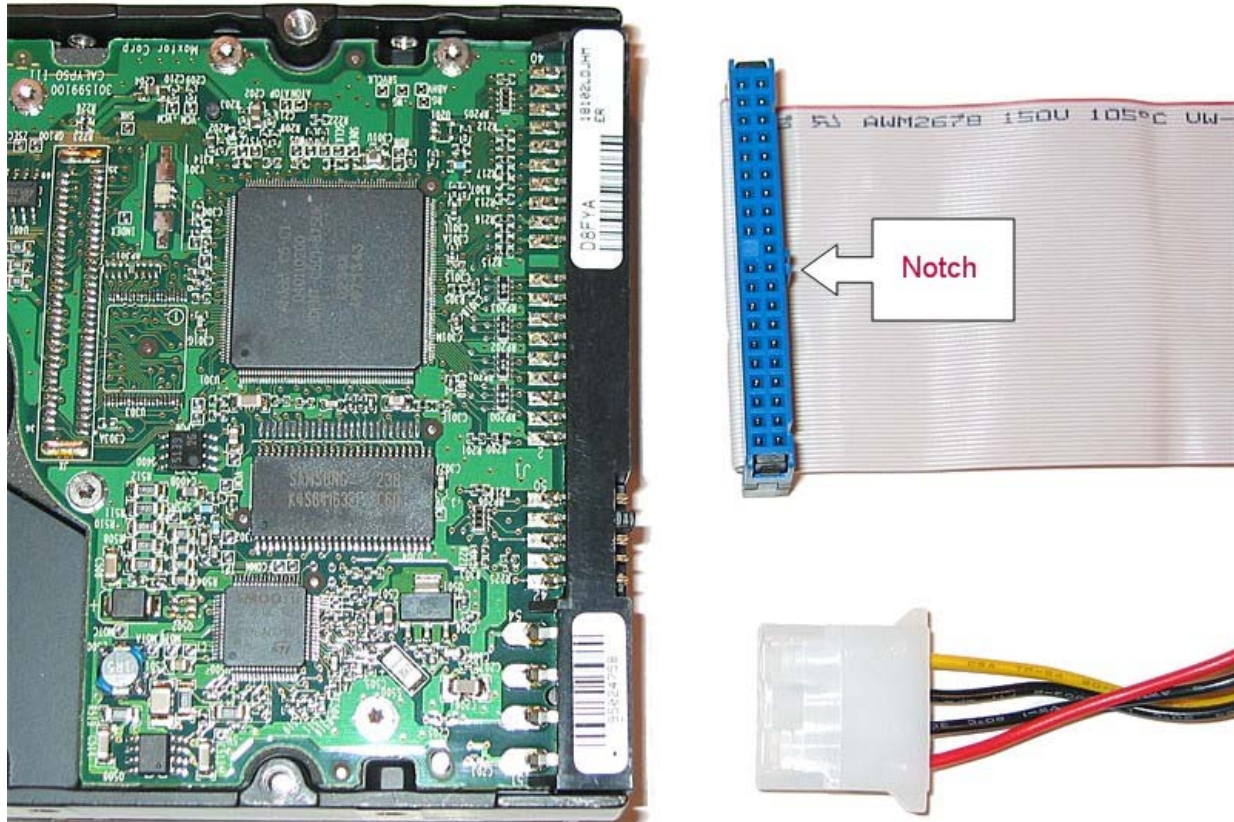



Figure 1.3: An IDE drive with data and power connectors.

 The notch must be oriented in the proper direction for the drive to work (it will fit on the wrong way in some configurations). Also of note is the bulk of the cable itself. This type of cable is not conducive to proper computer ventilation and can contribute to system overheating, causing hardware failures or erratic behavior.

SCSI

SCSI (pronounced skuzzy) disk drives have been available for decades. Its strengths include relatively fast throughput speeds, the ability to connect multiple disks to one disk interface, and automatic error recovery in most SCSI-enabled disks. SCSI has been popular for periods in the server, PC, and Macintosh segments of the computer industry.

Unfortunately, SCSI has many shortcomings. Primary among these is its constantly changing connectors and standards. Since the introduction of SCSI in 1986, it has undergone numerous revisions and updates. Nearly each update changes the connector configuration, requiring different cables and adapters to work properly. To get an idea of just how variable SCSI is, consider the following partial list of the major SCSI versions:

- SCSI-1
- SCSI-2
- SCSI-3
- Ultra-2
- Ultra-3
- Ultra-320
- Ultra-640
- iSCSI
- Serial SCSI

The complexity of the changing standards and different incompatible hardware revisions make management of SCSI devices difficult. For example, an existing investment of SCSI-3 cables and connectors is incompatible with new Serial SCSI investments.

It is also difficult for the IT professional to recognize the various SCSI connectors on sight, forcing most to carry references or refer to dedicated Web sites to identify hardware. For example, Figure 1.4 shows a diagram of a small subset of SCSI-1 and SCSI-2 connectors.

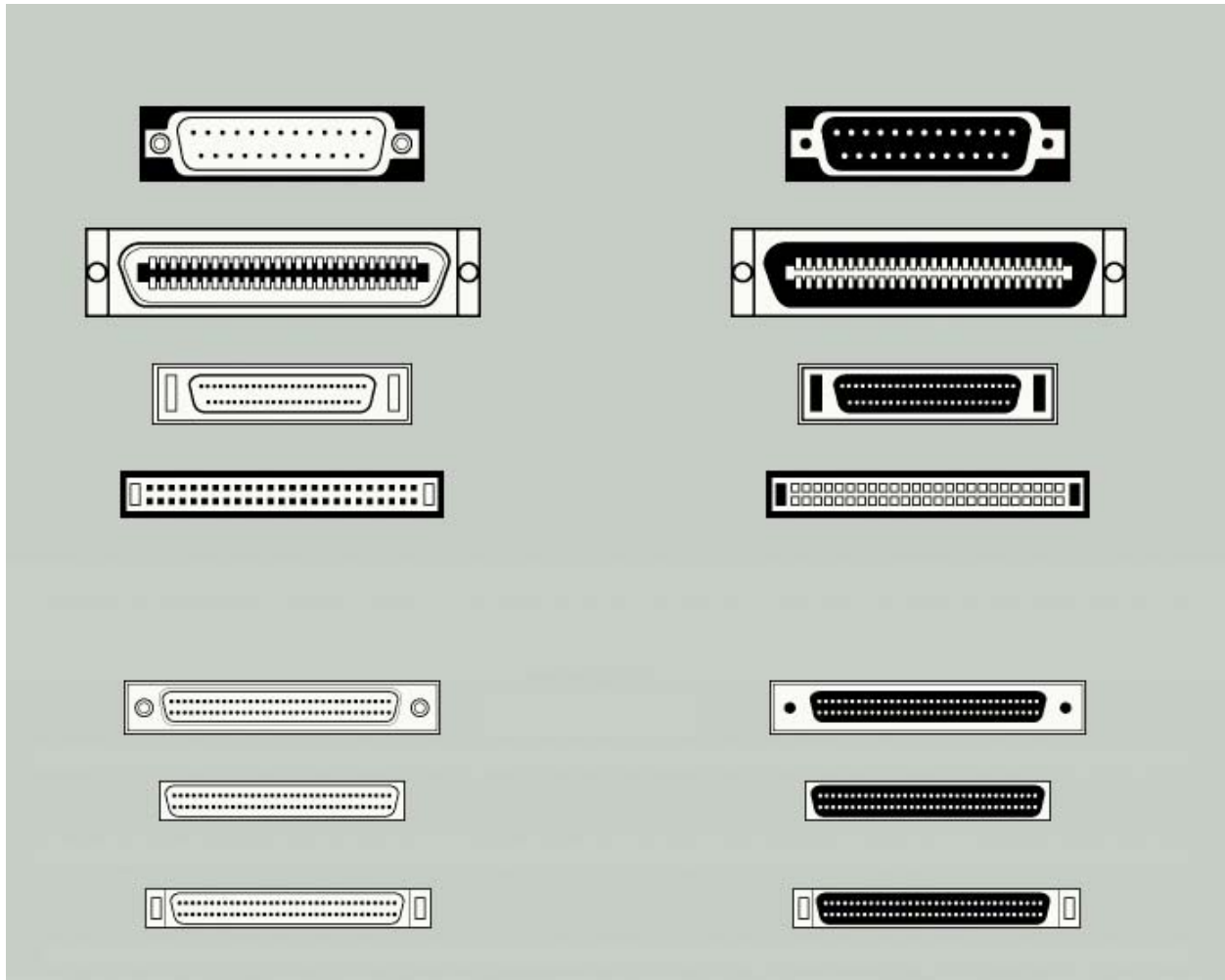


Figure 1.4: A sample of SCSI-1 and SCSI-2 connectors.

SCSI has also historically been a more expensive investment than other disk interfaces. Its complexity and requirement for advanced controller software often drive initial investment prices far beyond other similar technologies. Combine this with the fact that newer, simpler, and cheaper alternatives are available, and you'll understand why widespread use of SCSI-based devices is currently waning.

SATA

A newer evolution of IDE is SATA. It has evolved as hardware engineers examine the strengths and failures of previous interfaces. In this way, it has an advantage over all other standards because it can improve on weaknesses while continuing to build on strengths.

SATA has a much simpler and smaller connector than either IDE or SCSI. Although SATA connectors are often somewhat fragile plastic connectors, they are engineered to meet the needs of a normal volume of connections and disconnections. Figure 1.5 shows a typical SATA connector.



Figure 1.5: A SATA connector.

SATA was also designed to be cost effective. Both the interface electronics and cabling can be produced very inexpensively and can easily coexist with an IDE interface on the same computer. This reality has helped drive widespread adoption of the standard.

Another benefit of SATA is its greatly enhanced throughput and optimized data transmission. Typical SATA speeds begin at 1.5GBps, and newer standards are already in place (with similar hardware) that provide 3GBps throughput. Currently, most new high-end computers are equipped with SATA drives.

Disk Interface Wrap Up

The selection of a disk interface should be made on a cost-benefit basis. If the benefits of the more expensive formats outweigh the costs, that interface is the right one. You should also take into consideration current and future ownership costs, such as the cost of later disk expansion and the organization's current and future storage needs. However, none of these interfaces can be viewed as an absolutely "wrong" selection.

Fault-Tolerant Disk Systems

Computers are generally made up of numerous electrical circuits that show little or no wear over time. They don't wear out primarily because they don't move—the electricity moves, but the components do not show signs of wear from the electrical signals. However, unlike most other parts of a computer, hard disks contain numerous moving parts. These moving parts include disk platters spinning at thousands of revolutions per minutes and read/write heads traveling back and forth over the disk. These high-precision components are designed within very tight tolerances and generally can last for years. But they do wear out and fail more often than other computer components simply due to their design.

Designers identified this potential weakness very early in the evolution of disk storage technology. They devised a standard rating, the mean time before failure (MTBF), to describe how long a disk with constant power and normal usage should last before it fails. This rating is somewhat arbitrary because of its prediction-based methodology, but it does help systems administrators discriminate between drive features. It also reminds administrators that disks are prone to failure and that measures should be taken to mitigate this risk.

A very popular disk-failure risk mitigation technique is to configure more than one disk to store the same data. For example, if a customer database is stored on a single hard disk, when that disk fails, the data is inaccessible. However, if that same data is stored on three hard disks, it is highly unlikely that all three disks will fail at the same moment. One failure does not render the data inaccessible. This disk redundancy method became very popular beginning in the late 1980s and continues its popularity today.

In 1988, a specific scheme that uses multiple disks for data redundancy was defined. This scheme was called Redundant Array of Inexpensive Disks (RAID). RAID defines several levels of redundancy using a number of disk drives. What sets RAID apart is that in general the redundancy is handled by hardware, such as a RAID-compliant drive chassis or a RAID-enabled disk controller. The redundancy provided by hardware-specific solutions can be very fast and enable online disk recovery operations that would not be possible if an OS or other software-based solution was used.

Some of the popular RAID levels include RAID 0, 1, and 5. Many RAID schemes employ either these levels or a combination of these and other redundancy schemes. Understanding these three key schemes will help you understand RAID overall.

RAID 0

RAID 0 is also known as a striped set. This level of RAID writes data across exactly two physical disks with no redundancy or parity information. The loss of one disk results in the loss of all data, as there is no method for recovering data. For that reason, RAID 0 is actually not redundant at all and is not used where data redundancy is required. However, it is often mentioned as a RAID level. RAID 0 is frequently used in high-performance computer systems as a method to increase disk throughput speeds.

RAID 1

RAID 1 is a true redundant scheme. Also known as disk mirroring, it is used to ensure the integrity and availability of data by making an exact copy of all data. In this scheme, exactly two disk drives are used. The drives are maintained as exact copies (or mirrors) of each other at all times. If one disk fails, the other can continue to function alone while repairs are made. Because the likelihood of both drives failing at the same moment is remote, this scheme is considered useful and many systems employ it.

However, the cost-per-byte of a RAID 1 implementation is relatively high compared with other redundancy schemes. For example, two 500Gb drives configured as RAID 1 yield 500Gb of accessible space—half is available, the other half is for redundancy. With the cost of disk storage continuing to fall (at the time of this writing), this is not usually a cause for concern.

RAID 5

Although RAID 1 provides excellent data redundancy, it has a high cost per byte. RAID engineers looked for a way to reduce the cost of disk overhead while still providing redundancy (remember, this was done when disk drives were still very expensive). They came up with a scheme called RAID 5, also known as a striped set with parity.

In RAID 5, three or more disks are used. Data is written in blocks (stripes) across all but one of the disks. On the disk on which data is not written, parity information (also called a checksum) is stored. RAID 5 is often implemented in hardware in the form of smart disk controllers or smart RAID enclosures, so the computer and OS do not have to perform the complex parity calculations or disk management tasks.

When one disk in a RAID 5 array fails, the system continues as normal because the data from the lost disk can be calculated from the remaining data and parity information. System performance may temporarily decrease while that disk is down because of the extra operations performed, but that is more than made up for by the system uptime provided.

There are two big benefits of RAID 5. First, the failed disk can usually be replaced and initialized while the system is still online, virtually eliminating data downtime. Second, the cost per byte is much lower than that of RAID 1.

RAID Wrap Up

There are a number of schemes available to guard against the fallibility of modern disk drives. RAID schemes are very popular and are often implemented in hardware solutions that partially or completely abstract the OS from the RAID details. These schemes can prove effective in increasing uptime, but care should be given as to which scheme is implemented to ensure that the appropriate level of data redundancy is achieved.

File Systems

Disks store data using their own format, and the electrical connection between the disk and the computer has its own format. However, neither of these formats is conducive to easy use by application-level programmers or systems administrators because the formats are far too detailed and device-specific. Programmers and administrators need a way to logically organize, store, and retrieve data that is abstracted from the low-level mechanisms of data transmission and storage. File systems provide that layer of abstraction.

File systems are methods to store, organize, and retrieve data on a disk. They are often abstracted by the OS and made transparent to the user. For example, most Windows users cannot tell whether their file system is File Allocation Table (FAT), New Technology File System (NTFS), or High Performance File System (HPFS) unless they're looking for some specific feature only available on one of the systems.

There have been several significant file systems developed for the Windows platform. The most significant are FAT and NTFS. These file systems differ greatly in their capabilities and internal structure.

FAT

When MS-DOS was first being developed, Bill Gates needed a basic file system to store and retrieve data. His development efforts led to the first version of the file system he called FAT in 1977.

FAT is an uncomplicated file system and was very appropriate for the era in which it was created. It stores data in a very basic format because computers of those days didn't need a complex hierarchical or extensible file system. It takes up very little space for itself because disk space was at a premium. Many features simply weren't considered because they weren't part of the thought process: robustness, error recovery, extended file descriptors, and security being good examples. None of these features were intended to be in Windows, so the file system had no need to support them. The file system was also not extensible, because at that time there was no concept of changing or extending the data that the file systems supported.

Many current administrators feel that FAT is a useless technology and should never be used. Although it is true that FAT isn't as advanced as other modern file systems, it certainly has its place in today's environments. For example, FAT is almost always used on removable media such as floppy disks and thumb drives. You can also use FAT for backward compatibility with other OSs in dual-boot scenarios, such as when you need to use MS-DOS and Windows NT on the same single-disk system. FAT comes in three distinct variations: FAT12, FAT16, and FAT32. The difference is in the number of bits used in their data addressing: 12, 16, and 32, respectively.

FAT12

The oldest version of FAT is FAT12, which stores a maximum of 4077 files and supports up to a 32MB disk. Although this version was replaced by FAT16 for hard drive use as PC hard drives began to become available, FAT12 is still in use as the preferred format for floppy disks. Floppy disks have such limited space, and FAT12 can address it all with very limited overhead, making it an appropriate file system.

FAT16

FAT16 is nearly identical to FAT12 except for its use of 16 bits in its addressing scheme. But this minor architectural change allows FAT16 to address hard drives up to 2GB and store up to 65517 files. FAT16 was very popular with MS-DOS and versions of Windows up to Windows 98.

FAT32

In 1996, Microsoft recognized that hard drives were growing past the 2GB address limit of FAT16. The company addressed this problem by doubling the number of address bits to 32, creating a new file system called FAT32. This was first released in a service pack for Windows 95 and then Windows 98. This change allows FAT32 to manage hard drives of up to 2TB and store more than 200 million files. FAT32 is still in widespread use because it can manage current disk needs.

You do not need to know the detailed specifications of FAT. What you should remember is that FAT is in somewhat common use today. In general, disks that do not need to run FAT for a specific reason should be upgraded to NTFS eventually to get the numerous benefits of that advanced file system. But there are still several legitimate uses for FAT, and there is nothing fundamentally wrong with using it.

NTFS

When Microsoft was developing Windows NT, they recognized that FAT was not capable of much future growth. FAT had a number of design limitations and was not extensible. Thus, the software architects began to develop a new file system from scratch. The file system they designed was NTFS and it premiered in Windows NT 3.1.

NTFS was an enormous step forward. It had a number of integrated features, including:

- Ownership attributes
- Security descriptors
- Metadata support
- Atomic operation with transactional logging
- Support for volume sizes up to 16 exabytes
- Support for international languages
- Extensible attributes

Although all these features were enormously beneficial, one that bears further mention is extensible attributes. Essentially, this feature allows a software developer to customize NTFS in the future without having to redesign the entire file system. For example, when Microsoft integrated file system encryption in Windows 2000 (Win2K), the company simply extended the functionality of NTFS. Doing so avoids costly changes or upgrades for programs and prevents broken functionality.

Although FAT was designed as a list of files on a hard drive, NTFS was designed as an extensible database that could store, among other things, files and directories. Thus, NTFS can be extremely efficient despite storing enormous amounts of data. It also means that NTFS can organize free and used disk space rather easily. It will become clear how important this is later in this guide.

NTFS is the preferred file system on Windows-based computer systems today. It is the default file system for all new drives. You should consider using NTFS whenever possible.

Other PC File Systems

There have been other older file systems that were included with Windows in the past. In addition, many file systems have been ported to Windows over the years, including some that were never intended for use on a PC. Two are worth briefly mentioning, for very different reasons. One, HPFS, used to be supported in Windows NT and OS/2, so you might encounter it on rare occasions. The other, ext3, is not supported by any Windows version but is popular enough that you should be aware of its existence.

HPFS

HPFS was designed to work with OS/2. It had a number of advanced features and was the file system of choice for OS/2 users, and was the preferred file system for volumes between 200 and 400MB (as this was its optimal operating size). Full support for HPFS was included in Windows NT 3.1 and 3.5 to both support upgrades from OS/2 servers and to support POSIX-based applications that required access to an HPFS volume. However, lack of use of this feature prompted Microsoft to remove the ability to create HPFS volumes and then finally all support for the file system.

It is rare to encounter HPFS-enabled computer systems today. Unless there is a critical need for maintaining HPFS on a system (for example, a critical application requires it), consider converting the volume to NTFS or upgrading the OS to a more current version.

ext3

ext3 is the default file system for many Linux distributions. It is not officially supported on any Windows system. However, it is somewhat popular in the industry due to its inherent recoverability characteristics.

File System Wrap Up

There are a number of file systems available. Many are older, inefficient on large modern disk drives, and only suitable for limited situations such as backward compatibility. For most Windows-based computers, NTFS should be the file system of choice.

How Disks Are Used

At a very basic level, disks are written to and read from. But that level of understanding doesn't help you make decisions about how to manage storage. You need to probe a little deeper.

Let's take a look at a very common example. Suppose that SERVER01 is a Windows Server 2003 (WS2K3)-based file and print server in your company. On average, about 100 users have daily interaction with this server for data storage and retrieval and print jobs. SERVER01 is a top-of-the-line Xeon-based server with 8GB of memory and a 2TB disk array. The disk storage is configured as one non-fault-tolerant storage volume, and to address disaster recovery, nightly backups are made and sent offsite.

During an average work day, 400 print jobs are sent to SERVER01. The network bandwidth supports the data transfer from the clients just fine. When the print job is received by SERVER01, it is temporarily written to the disk array before being sent to the printer. Once the printer acknowledges receipt of the print job, the temporary file is deleted. This is the way printing works in Windows.

Also during the day, several hundred Microsoft Office files, such as Word documents and Excel spreadsheets, are accessed and edited on the server. Some files are just a few kilobytes in size, and others are quite large, as they contain graphics or video clips. During the normal operation of Microsoft Office software (and indeed most business software today), the files are saved to the server periodically as temporary files. These temporary files are placeholders to help recover an in-process document in the case of disconnection or a computer crash. It is not uncommon for tens or even hundreds of temporary files to exist on a heavily edited document. Once the file is saved and closed, all the temporary files it created are deleted.

In this small example, you can see that thousands of files are created, deleted, and edited throughout the course of a normal day on SERVER01. On the surface, this doesn't present a problem, as there is plenty of space and hard disk bandwidth. But if you look deeper, you'll see that there is the potential for significant stability and performance impact with this type of operation. Some of the disk-based performance considerations include fragmentation and I/O bandwidth.

Fragmentation

Data is normally written to and read from hard disks in a linear fashion—one long piece of data. This is done to optimize disk reading and writing and increase performance. When numerous files are written and deleted, gaps in the drive's free space will appear. These gaps will affect future files because those files must fit into the gaps. If the files don't exactly fit into one gap, it will have to be placed into two or more gaps. This is fragmentation.

Consider that a hard disk is a huge surface that just stores 1s and 0s. There is only one way to read those 1s and 0s. The disk read/write head must move directly over the data and read it. If all the data for a file is in one small area, the read/write head may need to move very little or not at all to read the whole file. But if the data is scattered all over the disk, the read/write head needs to move a great deal to gather the data for the file. This difference can be negligible on small isolated files or an infrequently used system, but on a file server with thousands of files, the lag can quickly add up to a noticeable performance decrease.



One way to think of this is that the Earth is covered in tiny pebbles, and each pebble has a black side and a white side. Each pebble represents a bit of storage on a hard disk. The read/write head is a helicopter. Whenever you need data, the helicopter flies to the part of the Earth that has the file, lands, and the pilot reads out the position of each pebble to the requestor. When you're writing a file, the pilot must fly to the proper position, land, and begin flipping pebbles to their proper position according to the data being written. So as you can guess, having all the pebbles necessary for an operation together in one spot would save a lot of flying and landing.

Two common misconceptions about disk fragmentation are that newly installed computers are fragmentation-free and that NTFS eliminates fragmentation. The first idea, that new computers are unfragmented, is simply untrue. New computers can have extensive disk fragmentation. Often this is caused by numerous writes and deletes during computer installation. On computers upgraded from a previous OS, the problem is exacerbated because the drive may have been fragmented even before the upgrade began.

Although NTFS does actively seek to avoid fragmentation, it can only do so on a best-effort basis. If there is a readily available contiguous extent to store a new file, NTFS prefers that over a fragmented extent. But such extents are not always available and NTFS will provide any available disk space for storage, including fragmented space. There is no built-in functionality for NTFS to defragment or ensure the contiguous writing of files.

Disk Bandwidth

One way to think of a computer is as a central data processor that reads and writes data. In that case, there are three performance considerations:

- How fast can I read data?
- How fast can I process the data?
- How fast can I write the data?

Most of the biggest computer breakthroughs and sales campaigns over the past several years have revolved around the processing consideration. The battle between Intel, AMD, and Motorola primarily revolved around processing power, not data bandwidth, because usually the different processor manufacturers can all use the same disk interfaces. Thus, as the computers beef up in the processor area, the data pathway becomes even more important.

Reading and writing data to RAM is relatively fast, as those data pathways are short and have few boundaries. But data access on long-term storage, such as hard disks, is different. The data must be converted to a different transmission method via a disk bus, such as IDE, SCSI, or SATA. This conversion takes a significant amount of time and resources compared with data access from RAM or a cache. Therefore, the disk access often becomes the performance bottleneck of a system.

Summary

There are several factors that go into disk management. Disk interfaces, file systems, OSs, and other factors all play a part in disk performance and reliability. Selecting the right combination of these factors can play a key part in the behavior of your system. But no matter which selections you make, it is likely that you'll need to understand long-term disk management and maintenance issues. Chapter 2 will discuss those issues in detail.

Additional Resources

For a more complete examination of disk architecture in Windows, see *Magnetic Storage Handbook* (McGraw-Hill Professional) by C. Denis Mee and Eric D. Daniel.

For more information about file systems, see *Windows NT File System Internals* (O'Reilly) by Rajeev Nagar or *Microsoft Windows Internals, Fourth Edition* (Microsoft Press) by Mark Russinovich and David Solomon.

For more information about fragmentation, see *How File Fragmentation Occurs On Windows XP / Windows Server 2003* at <http://files.diskeeper.com/pdf/HowFileFragmentationOccursonWindowsXP.pdf>. Chapters 2, 3, and 4 will also provide information about fragmentation.

Chapter 2: Issues with Disk Fragmentation

Chapter 1 explored how disks work. They were designed as efficient long-term data storage devices, and they've lived up to that design criteria well. The first disks were large, clunky, fragile, and had very limited storage capacity. Over time, disks have significantly evolved. A disk today might fit on a postage stamp, draw virtually no power, have a lifetime measured in decades, and have the capacity to store the entire Library of Congress. Performance has also come a long way with today's disk throughput being orders of magnitude more than even a decade ago.

Cost has always been a concern about disks. In Chapter 1, we learned that disks used to be extremely expensive and hence very rare. Today they're virtually commodity items. You can buy a reliable, high-capacity disk drive at any office supply store for less than the cost of a nice chair.

Overall the disk storage market has boomed and products are keeping up with demand. As an example of the drastic evolution in the market, at the time of this writing, a fully redundant disk array that provides one terabyte of storage can be implemented for less than \$1000 using off-the-shelf hardware and does not require specialized knowledge or extensive consulting. Such disk arrays were scarcely available to consumers and small businesses even 5 years ago and, when available, required extensive consulting with storage experts, specialized hardware implementations, and cost tens of thousands of dollars or more. In short, disk-based storage is getting cheaper, easier, and more commonplace.

Disk operation is not all paradise, though. There are many issues to consider when operating disks. None of them should prevent you from using disk storage. However, they should be taken into account when implementing and operating any disk-reliant system. These issues can include:

- Disk lifetime—How long will each disk drive work before it fails?
- Throughput—How quickly is data getting from the storage system to the computer?
- Redundancy—Is the system truly redundant and fault tolerant?
- Fragmentation—Is the disk system operating at optimum efficiency?

This chapter explores the most common issue in disk operation—fragmentation. It happens to all disks on all operating systems (OSs). It can affect the health of the system. And it's easily repairable.

Negative Impacts of Disk Fragmentation

Chapter 1 explored the cause of and provided a detailed explanation for disk fragmentation. To briefly recap, fragmentation occurs when data or free space on a disk drive is noncontiguous. There are a variety of causes for disk fragmentation, including normal use of disk storage. Although most modern systems attempt to prevent disk fragmentation, it is an eventual state for all systems. In this respect, disk fragmentation is akin to soapy buildup in a bathtub. No matter how much you rinse, eventually the soap will build up to noticeable levels. And like soap buildup, it can be fixed.

This chapter will explore the three main concerns that result from disk fragmentation:

- Performance
- Impact to data backup and restore operations
- Concerns for reliability and stability

For each of these concerns, we'll explore the root cause based on the understanding of disk operations that were established in Chapter 1. We'll then analyze the measurable result of disk fragmentation within these concerns. And during this analysis, we'll debunk a number of common myths about disk fragmentation. These myths often lead to misunderstandings of how fragmentation impacts a system. As a result, many administrators blame fragmentation erroneously for a whole host of issues, and many issues that go otherwise explained can easily be attributed to fragmentation with the knowledge provided here.


Performance

When an important computer completely stops working, it couldn't be more obvious. Users scream, administrators exert themselves, technical support engages, and management gets involved. In extreme cases, computer downtime can affect stock prices or make the difference between a profitable or unprofitable company. Most large organizations go so far as to assess the risk to their main systems in terms of dollars per minute of downtime. For example, a large airline might lose \$100,000 each minute their reservation system is down. This directly calculates to the company's financial success: if that system is down for 10 minutes, it could affect the stock price; if it's down for a day, the company could fold.

What happens when that same \$100,000 per minute system is 10% slower than it was last month? Do the same users scream or administrators feverishly attempt to address the issue? Do stockholders complain that they're losing \$10,000 per minute? No. Usually very little happens. Few organizations perform impact analysis on a partial loss of a system. After all, if the system is up, reservations are still being accepted. But consider that this 10% slowdown equates to measurable lost productivity. A slower system has extensive impact including fewer customers served per hour, less productive employees, and more burdened systems. This loss of efficiency could severely impact the business if it continues for a long time.

Most network and systems administrators establish baselines to help identify when this type of slowdown occurs. They usually watch for symptoms such as network saturation and server CPU utilization. These are great early indicators of a variety of problems, but they miss one of the most prevalent causes of system slowdown: disk fragmentation.

If your organization follows the Control Objectives for Information and related Technology (COBIT) framework for its IT processes and best practices, you'll quickly realize that defragmentation most cleanly maps to the DS3: Manage Capacity & Performance objective within the Delivery and Support domain. It can be argued that defragmentation can also sometimes map to the Ensure Continuous Service objective, but the most common fragmentation-related operational work falls under Manage Capacity & Performance.

 For more information about COBIT, see the ISACA Web site at <http://www.isaca.org/>.

There are a number of variables that must be taken into account when analyzing the impact of fragmentation on performance. For example:

- Some software does not interact with the disk extensively. This type of software may be slower to load initially but may be unaffected by fragmentation when fully running.
- Frequently, large software packages load a small subset of code when launched and then perform “lazy reads” during operation to continue loading the software. Disk fragmentation has a limited effect in this case because the software is designed to use disk throughput without impacting the user experience.
- If small files are used extensively, there may be little difference between fragmented and non-fragmented disks. This is because the files may be too small for fragmentation to have any effect. Often applications that use numerous small files perform extensive disk I/O no matter the fragmentation situation.

The easiest way to illustrate the effects of disk fragmentation is to take a look at several common examples of how disk fragmentation affects average computer systems.

Common Fragmentation Scenarios

As Chapter 1 discussed, the normal use of a computer system will inevitably lead to some level of disk fragmentation and a resultant decrease in system performance. However, there are a number of scenarios that are more likely than others to both cause and be impacted by disk fragmentation. Let's examine a few real-world scenarios.

Newly Set Up Computer

Every computer starts its existence as a newly installed and configured system. When you first take the computer out of the box, it has an initial setup already configured on the hard disk. Usually it comes preinstalled with Microsoft Windows and a number of applications. Some organizations customize this initial configuration by adding new applications and removing superfluous ones. Other organizations might install a preconfigured system image or reinstall a different OS. But the result is always a new OS with all the tools that the user needs to be productive.

During setup, many files are created and deleted on the hard disk. Part of a normal installation process is the creation of temporary files, often very large ones, and then the removal of those files at the end of the installation. As a result, the disk can be very fragmented right after initial system setup.

Consider the installation of Windows XP Professional and Microsoft Office. These are very common tasks in any organization and in most homes. During a typical installation of both of these software packages, approximately 473 files are fragmented with 2308 excessive file fragments (“The Impact of Disk Fragmentation” by Joe Kinsella, page 7). Other operations, such as applying current service packs and security patches, can exacerbate the level of fragmentation on the disk. As a result, a system can easily start its life cycle with far more fragmentation than expected.

Computer that Has Been Running a Long Time

Modern computers are built to be useful for several years. That is a very good thing from a return on investment (ROI) viewpoint. You want your expensive computer systems to work as long as possible.

However, over time systems often become slow. One of the primary complaints of computer users is, “My computer is so much slower than it used to be.” Certainly as new technologies come out newer computers can be much faster, often making older computers seem slow. But why should a computer slow down over time? There are a number of potential causes for gradual performance degradation, including:

- Larger applications consuming more system resources
- More software installed and running
- Increased workload as the user becomes more experienced
- Malware infections consuming system resources
- Disk fragmentation over time causing disk throughput slowdown

Any one of these can significantly impact a system’s performance. Often most or all of these elements affect an older system. But the last one, disk fragmentation, is often overlooked by systems administrators trying to regain lost performance. Heavy disk fragmentation, which naturally occurs over time, can easily decrease system performance by a noticeable amount.

Specific statistics for exactly what impact fragmentation has on specific applications are contained in the article “The Impact of Disk Fragmentation” by Joe Kinsella. In this article, a number of applications were tested both with and without significant disk fragmentation. In all cases, the system performed worse with fragmentation. For example, Microsoft Word was approximately 90% slower when saving a 30MB document with disk fragmentation than without. And Grisoft’s AVG took 215.5 seconds to scan a 500MB My Documents folder for viruses when the disk was fragmented compared with 48.9 seconds without fragmentation.

File Server

This scenario is the easiest to describe and the most common. One of the most common roles that a server holds is as a centralized file server. This role has been around since the earliest days of client-server networking. Users are comfortable with the concept of storing files on a central server to provide access to multiple users from multiple locations and to ensure that the files reside on a reliable and backed up store.

Many services provide expanded functionality beyond the traditional file server. Microsoft SharePoint Server, for example, combines a file server with a Web-based portal and rich communications tools. Within the context of this guide, any server that stores and provides access to numerous files is considered a file server.

During normal use, a file server will have a number of files stored on the local disk. A very simplified example of these files is shown in Figure 2.1, in which the user's application is storing several small files, all on the file server. After five separate file write operations, you can see that the disk is beginning to fill up, but because the file system attempts to use contiguous file space you see no evidence of fragmentation.

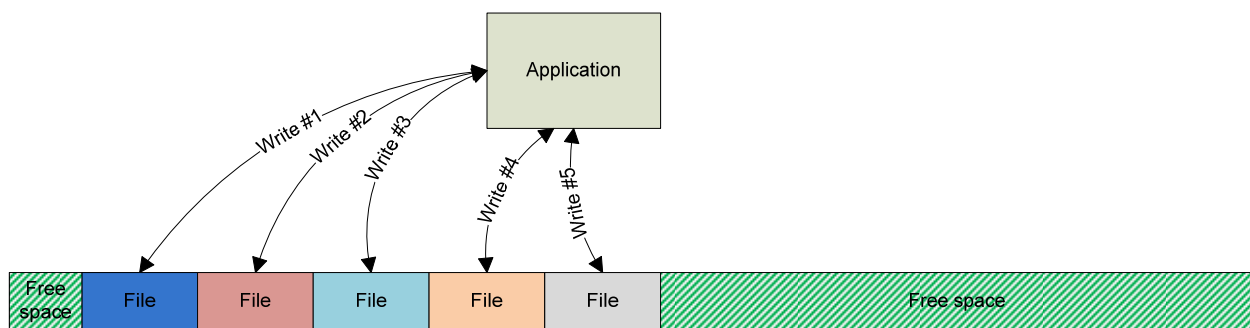


Figure 2.1: An application creating a number of small contiguous files on a hard disk.

After some amount of normal usage, the user will delete, rewrite, and add files to the file share. Figure 2.2 shows a typical example of what the disk might look like over time.

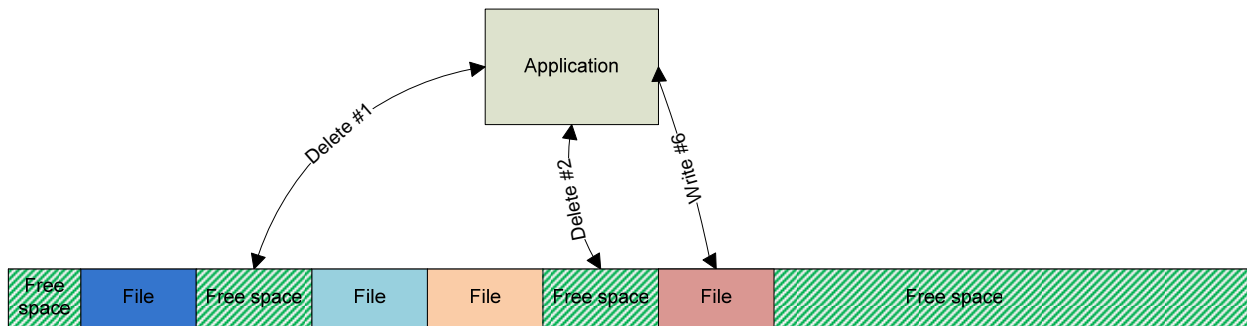


Figure 2.2: An application during normal operation, deleting some files, updating others, and writing new ones.

Notice in this diagram that there is now a significant amount of free space in separate locations. The files on the disk remain contiguous because when new files were added, as in write #6, there was still enough contiguous space to create contiguous file. Remember that the file system will *try* to use contiguous space first to avoid fragmentation. But because of system limitations (which we'll explore later in this chapter) this often doesn't happen.

Suppose the user stores a large file on the file server in the write #7 operation. This file is too large to fit in the one remaining contiguous extent of free space. Therefore, the system must split up the file wherever it will fit. The file, stored in its fragmented form, is shown in black in Figure 2.3.

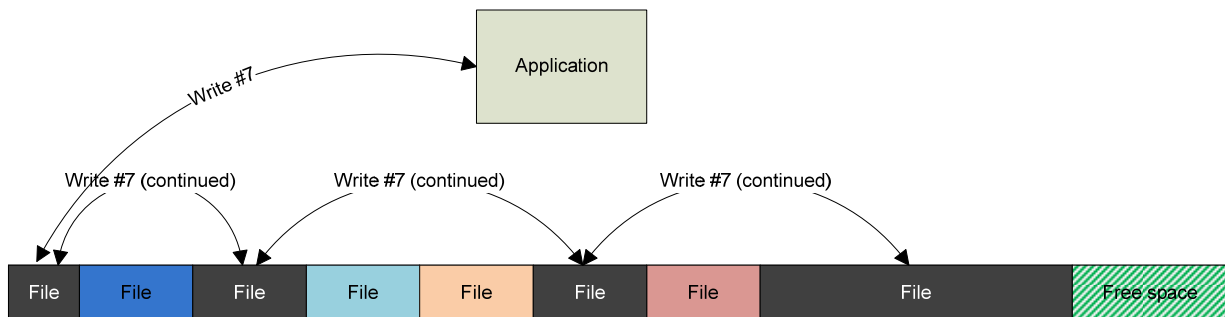


Figure 2.3: Writing a big file to the hard disk while the free space is fragmented.

This fragmentation is a fairly typical example of the type of operation that can happen hundreds or thousands of times per minute on a busy file server. As you might surmise, the problem gets even worse as the free disk space shrinks. Less free space means that the system must use any free space fragments, no matter where they are. This suboptimal condition can result in a thoroughly fragmented system.

Fragmentation can profoundly affect a file server. The core function of a file server is to read and write files and communicate the data from those files on the network. Any slowdown in reading or writing the files will have a negative impact on the system's performance. This situation can also shorten the life of disk drives by causing them to move the disk heads more often to read and write fragmented files. Although there are no extensive studies on this case, it makes sense that more work for the drive means a shorter life.

Computer with a Full Hard Disk

Although very similar to some other scenarios, this one is distinct in its root cause. The condition occurs when a system's hard disk becomes full over time. This happens on many computers because drives can fill up with downloaded content, new data such as photos or scans, or new applications that require more hard disk space. Inevitably, most systems come to a state where the hard disk has very little free space left.

Fragmentation is almost guaranteed to be a problem when there is little free space left. The OS is scrambling to use the available free space no matter where it's located. When a new file is written to the disk, it will probably be fragmented. Figure 2.4 illustrates the point that the application cannot always write a contiguous file when free space is scarce. If the application needs to create a file the same size as the red file in this figure, it will have to use at least two separate free space allocations.

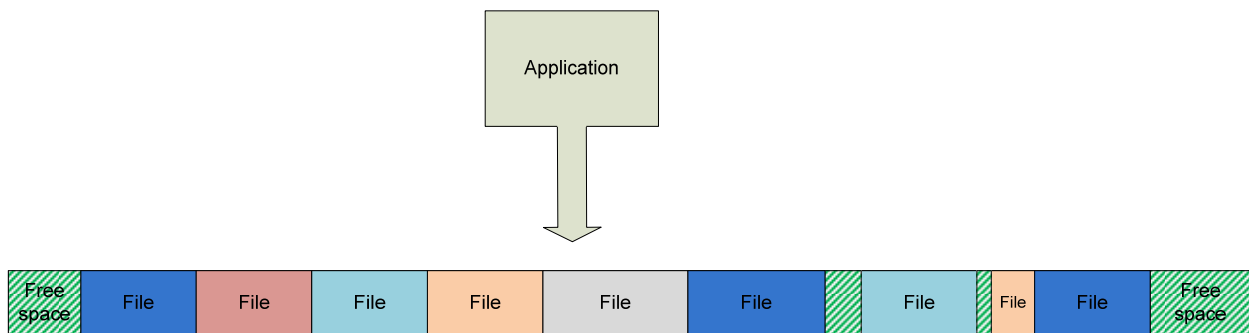


Figure 2.4: *With such sparse free space, the application will almost certainly create a fragmented file.*

Compounding the problem is that the most common fix, defragmenting the disk, will probably fail. Virtually all defragmentation software performs the work by taking fragmented files and writing them as a contiguous file to another spot on the hard disk. If there is not enough space to create a contiguous spot for the new file, it cannot be defragmented. That is the reason most defragmentation software packages alert the administrator when free space gets low enough to cause a problem.

Data Backup and Restore

The industrial revolution moved the focus of labor from people to machines. The entire economic landscape of the world was changed as factories began to develop and produce products on a massive scale. Steam power, iron work, transportation advances—the world changed significantly. Many authorities agree that the world is currently undergoing another revolution. This one is not about steam power or making a better factory. This one is about information. Information is considered the new economic focus.

Many modern industries are data-centric. Consider that many jobs deal only with data and its value: computer programmer, network manager, or even Chief Information Officer (CIO). Some industries exist entirely on data, such as Internet search or advertising placement. Consider where Google, Microsoft, or Doubleclick would be without the data that they've invested enormous amounts of money and time to develop. To these industries, their data is just as important as grain is to a farmer or the secret recipe is to Coca Cola.

Data Backup

Companies that place value on their data go to great lengths to protect it against loss or compromise. They cannot lose the central focus of their business when a hard disk fails or a power supply blows up. These companies invest in a variety of data protection methods to minimize loss and downtime. One of the most basic and most effective methods is data backup.

In short, data backup is the process of copying data from a main source to a secondary source; for example, burning a DVD with a copy of a customer database stored on a server's hard drive. Normally, the data backup is carefully stored in a secure location so that when the original source is compromised, it is likely that the backup will be unaffected.

Data backup is often a slow process. There are several factors that contribute to data backup being slow:

- Volume of data can be enormous
- Inability to take the data offline, requiring a complex backup scheme to copy the data while it's being changed
- Scarce system resources available on data host
- Fragmented state of data source

Most standard data backup practices have built-in mitigations to these factors. They include scheduling backups during periods of system inactivity, purging unwanted data before the backup begins, and (less frequently) scheduling system downtime to coincide with data backup. However, many organizations ignore data fragmentation as a component of data backup. It's simply not part of their thought process. This is an incorrect oversight.

Data fragmentation can significantly impact a backup process. As we've already seen, fragmentation leads to delays in reading data from the hard disk. Data backups rely on reading data as quickly as possible for two reasons: to speed the backup process and to efficiently supply data to continuous-write devices such as DVD drives. Heavily fragmented data will take longer to read from the disk. Thus, at best, the backup process takes longer to complete. The worst case is that the backup will fail due to the delay in supplying data to the continuous-write data backup device.

The amount of impact that disk fragmentation has to disk backup depends greatly on the destination of the data. We'll look at four types of backup destination schemes within the fragmentation context: disk to tape (D2T), disk to disk (D2D), disk to disk to tape (D2D2T), and disk to optical (D2O).

Disk to Tape

When disks were first being used, they were terribly expensive. Costs of hundreds or even thousands of dollars per megabyte were common. The online storage that disks provided at the time were novel and created new opportunities for computers, but a solution had to arise to mitigate the fact that these disks were expensive and provided limited storage. Fortunately, a solution was readily available.

Tape-based storage had already been around for some time. Tapes were cheap, removable, and easily storable at multiple locations. This provided an economical, scalable storage option. Offsite tape storage added the benefit of disaster preparedness. This copying or moving of data from disk to tape storage became known simply as D2T and has been the most widely used backup scheme for decades.

D2T is partially affected by fragmentation because the disk read operations from the source might be delayed due to excessive disk fragmentation. If the system is already I/O constrained, fragmentation could have a significant effect on backup performance. Tape is also an inherently slow backup media because of its linear nature and because removable magnetic media cannot handle the same throughput as dedicated media. To overcome this shortcoming, the D2D and D2O schemes emerged.

Disk to Disk

Disk drive systems are currently the fastest primary storage systems available. Chapter 1 concluded that disk throughput has significantly increased as storage capacity has gone up. Truly amazing amounts of data can be written to disk in time so short it may not even be noticeable. And disk storage has become less expensive with each passing year. It's still more expensive than tape or optical media, however.

When speed of backup is the most important element in deciding a backup scheme, most systems administrators go with a D2D solution. Thus, the data is copied or moved from the primary disk to another disk, designated as the backup disk. The backup disk could be connected to the system by a variety of means such as Universal Serial Bus (USB), Small Computer Systems Interface (SCSI), or IEEE 1394 Firewire, or in some cases, by network connection (although this is slower). The disk obviously needs to be as big as or bigger than the original data being backed up. Ideally, the backup disk is large enough to store the backup data for several computers or servers to improve the efficiency of long-term data storage.

D2D backup is very sensitive to disk fragmentation. Both the source and the backup disk can become fragmented. In particular, the backup disk can become very fragmented due to the enormous volume of data it is storing and calling up as part of the backup process. Because fragmentation can occur both at the source and backup points, both sides can slow the process and together the process can be significantly affected.

Disk to Disk to Tape

D2D is quick but expensive. D2T is slow but cost effective. A good compromise between these two schemes is to initially back up data to a disk, then later move the backup data to a tape. This method is called D2D2T backup and is illustrated in Figure 2.5.

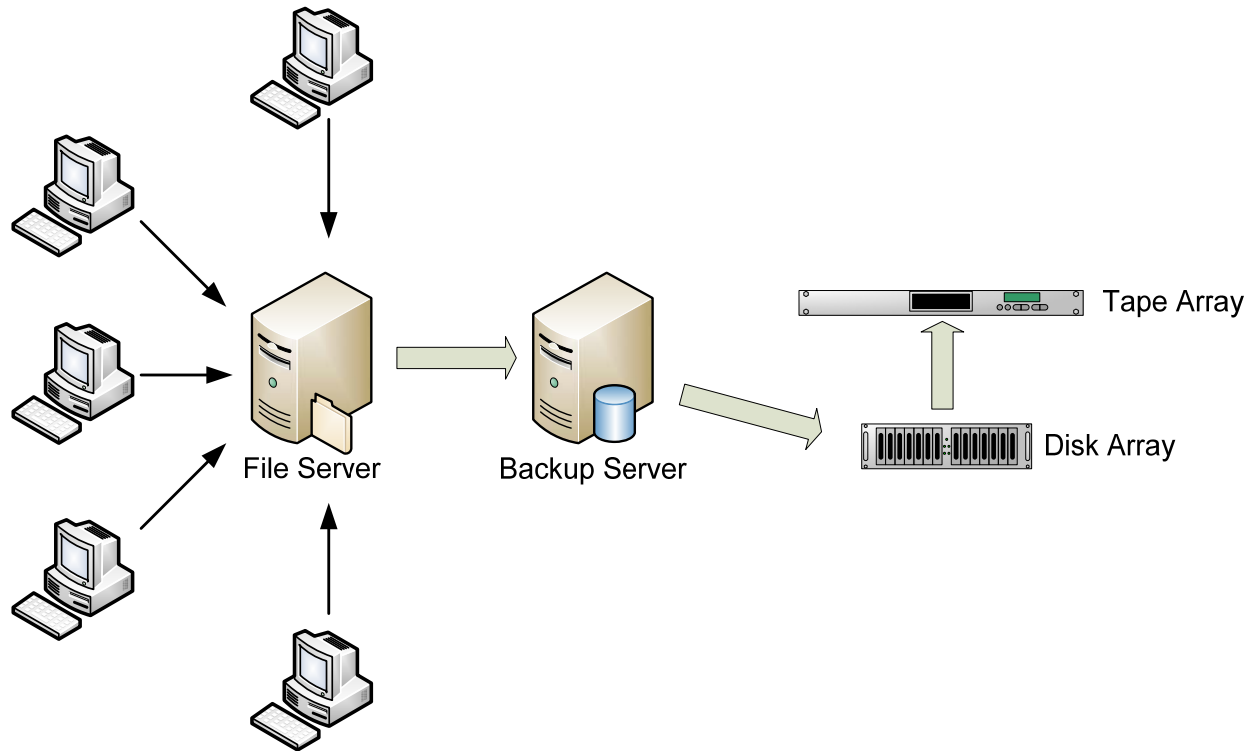


Figure 2.5: The data flow for a D2D2T backup.

The example in the graphic shows a frequently used file server. Performance for a D2T backup would significantly impact the users. Instead, a D2D backup is performed first. This can be done very quickly with little impact to the server's performance. Once the data is on the backup server, it can be written to tape more slowly without impacting the users.

This efficient and cost-effective solution has the same fragmentation concerns as D2D. To be most effective and have the least user impact, the disk drives should be defragmented.

Disk to Optical

With the recent explosion of inexpensive, high-capacity optical media (for example, DVD+R, DVD-DL, and so on) the D2O backup method has become a realistic option. Backing up to optical media can be very fast compared with the D2T backup method and the disks can be destroyed and replaced whenever a new backup is conducted. The disks are also very easy to send to an offsite storage facility for redundancy because they're both lightweight and durable. Although D2D2T is a very popular option for enterprise-wide backup today, D2O will probably gain market share in the coming years.

As mentioned previously, writing to optical media is very dependent on timing. If the data is not ready to go to the optical disk at the right moment, a buffer underrun will occur. This same risk applies to D2O backup. Although there are many potential causes for buffer underrun conditions, one of the principle ones is disk fragmentation. Any delay in reading the data for presentation to the optical media could cause the underrun and ruin the backup attempt. Luckily, disk defragmentation can help avoid such failures.

Data Restore

The whole purpose of data backup is to provide a way to restore data in the case of catastrophic loss. Usually, the backup software does not take any specific actions around how the data is written when it is restored. That is not its job. The backup software simply writes files to the hard disk. This can cause a problem.

When data is read back from its backup location to a system for use, it is usually written to the system's hard disk to ensure maximum performance (as opposed to accessing the data directly on the backup media). Very often, the data is fragmented when it is loaded from the backup media to the system. That is because it is usually an enormous amount of data being written to disk at one time and must necessarily use up a great deal of the system's disk storage. Figure 2.6 shows this happening for one large file.

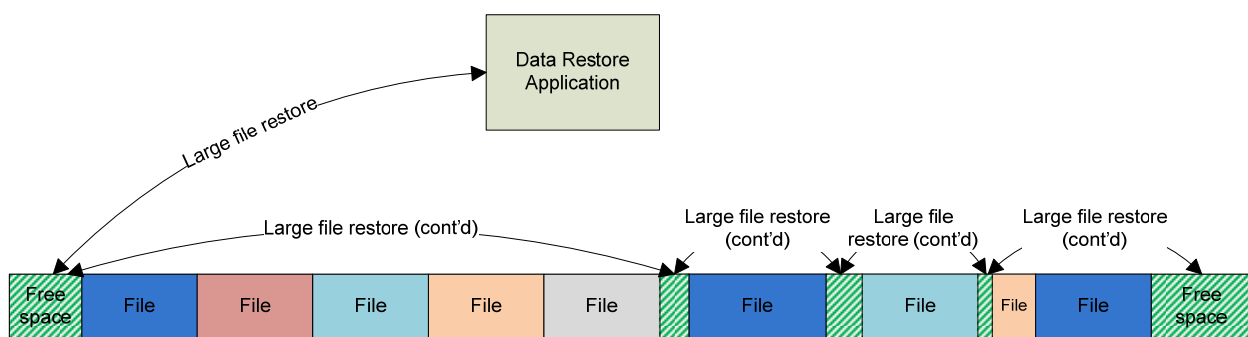


Figure 2.6: Restoring a file to a fragmented disk is slow and results in more fragmentation.

Unless the system has contiguous free space available that is much larger than the size of the backup, the data will probably be fragmented when it is written.

Stability

Most systems administrators consider that, at worst, disk defragmentation is a minor inconvenience. Even those who do understand it to some extent believe that fragmentation has a very limited effect on the system and that in most cases it is unnoticeable. We've already discussed that the performance difference can be very serious, causing noticeable system degradation and loss of efficiency. Let's take a look at how fragmentation affects system stability.

The OS, applications, and data for most computer systems are stored as files on the hard disk. When a computer is turned on, it first checks to ensure that the system's components are functioning properly. It then loads the first OS files and transfers control of the computer to the OS to complete its load. Usually, this process completes very quickly and the system is up and running.

What happens to a system when the core OS and application files are heavily fragmented can be surprising. Some examples of problems that can occur when these files are fragmented include boot failure, program and process failure, media recording failure, premature hardware failure, and memory-based system instability.

Boot Failure

We examined the possibility of performance degradation earlier in this guide. However, it is possible that the system can go beyond slowdown to failure. Although rare, it is a possibility. The reason is that during OS boot, key system files must be loaded in a timely manner.

The most likely cause for this scenario can be traced back to a heavily fragmented master file table (MFT) on a Windows computer running the NTFS file system. The MFT holds key information about all other files on the hard disk and contains the map of free drive space. Fragmentation of this file has a cascade effect, causing all other disk input/output to slow down while the MFT is read. Windows attempts to keep the MFT in a single extent but often fails to do so, especially when using a small or nearly full hard disk. Although other key Windows files can cause boot failures if they're fragmented, the MFT usually has the biggest impact.

If the key OS files are not loaded within a predetermined time limit, the OS may think that it has become compromised or corrupted. In that case, it will display an error message and stop the boot sequence in order to protect system integrity. This results in system downtime and could potentially require reinstallation of the OS to fix (unless you have a solution in place to defragment offline systems).

Program and Process Failure

Similar to OS boot failure, programs and processes can fail for the same reasons—slow load times causing timeouts and errors. Particularly sensitive to this type of failure are database applications or any network-based application that requires a specific level of responsiveness. The disk fragmentation can sometimes impact performance to the point that these applications cannot communicate quickly enough and they fail.

Programs can fail because their own files are fragmented and they take too long to be read from disk. Failure can also occur when the program is large enough to force the system to use virtual memory. When this occurs, the OS temporarily writes memory data to the pagefile, a file on the disk specifically designated for this type of data. If the pagefile is also fragmented, the slowdown is compounded and can cause system-wide program failure due to its resource consumption.

The instability caused by program failure is exacerbated on systems that have a small amount of memory. These systems are likely to write memory to the pagefile more quickly than systems with enormous amounts of RAM. Because these low-memory systems are already challenged for performance, having a fragmented disk will cause an even greater system slowdown and potentially application or OS instability.

Media Recording Failure

Optical media recording (for example, CD, DVD) requires that the data be fed to it in a continuous stream. Any significant slowdown or stoppage in the data stream to the recording can cause the entire recording to fail. To help prevent this condition, most optical drives have a buffer that they use for the data stream when the data flow slows down or temporarily stops. When the buffer is used and there is still not enough data to continue writing the media, a buffer underrun event occurs and the optical media is rendered useless. When the disk drive is fragmented, the CD or DVD burning software may not be able to retrieve data quickly enough and the write could fail.

Premature Hardware Failure

Chapter 1 explored how disk drives work. We know that when the disk is read, the read heads must move to the appropriate spot on the hard disk, read the data, and then move to the next spot on the hard disk. Consider that fragmentation is the state when one file resides in more than one noncontiguous place on the hard disk. In that state, the read heads must move to several spots on the hard drive to read a single fragmented file. On a system that conducts intense file I/O (for example, a file server or a heavily used desktop computer), there could be hundreds of fragments that all require the repositioning of the read heads to capture a single file.

All that movement has a negative impact on the disk's longevity. Because the disk is a mechanical device, each movement affects the device's life. If disk access is optimized, the mechanical component of the device is likely to last much longer than one that has to work much harder to accomplish the same task. If the read heads have to move violently each time a read or write request is processed, the extra effort could have a negative long-term effect.

You should not consider a computer system that has fragmented files to be severely in danger. This condition is not an imminent threat to the hardware's immediate future. But you should consider this to be a long-term risk that can be easily mitigated.

Memory-Based System Instability

Earlier, we considered what could happen when the pagefile becomes fragmented. Programs and processes could fail to load or operate correctly. This is obviously a cause of concern. However, there is another symptom that comes from the same root cause.

In situations of heavy fragmentation, the OS itself could generate errors or potentially shut down. The root cause is similar to the earlier section in which the slowdown of access to the pagefile causes timeouts and errors reading files. The OS could interpret this condition as being out of memory, as the pagefile is seen as an extension of system memory. When the system is out of memory, everything slows to a crawl and any number of errors can take place. At that point, the system is unstable because system services and processes may shut down when they're unable to access memory.

Summary

There are several problems that result from disk fragmentation. The most commonly understood problem, that of performance degradation, is certainly the most likely to occur on most systems. However, there are a number of other serious problems that can come up because of disk fragmentation. They range from the inability to write optical media all the way to system instability and crashes. You should be aware of these issues when examining unstable or nonperforming systems so that you recognize the symptoms of a heavily fragmented disk.

One important point that we did not cover in this chapter is what to do about fragmentation. You can see that it is a bad thing for your systems, but you don't quite understand how to fix the problem. Should you delete files from the disk? Should you run the built-in Windows defragmenter? Should you go buy a defragmentation solution? We'll examine all of these options in Chapter 3 so that you can make the decision that works best for your environment.

Chapter 3: Solving Disk Fragmentation Issues

Chapter 1 explored how disks work. They were designed as efficient long-term data storage devices, and they've lived up to that design criteria well. The first disks were large, clunky, fragile, and had very limited storage capacity. Over time, disks have significantly evolved. A disk today might fit on a postage stamp, draw virtually no power, have a lifetime measured in decades, and have the capacity to store the entire Library of Congress. Performance has also come a long way with today's disk throughput being orders of magnitude more than even a decade ago.

Cost has always been a concern about disks. In Chapter 1, we learned that disks used to be extremely expensive and hence very rare. Today they're virtually commodity items. You can buy a reliable, high-capacity disk drive at any office supply store for less than the cost of a nice chair. Overall the disk storage market has boomed and products are keeping up with demand. As an example of the drastic evolution in the market, at the time of this writing, a fully redundant disk array that provides one terabyte of storage can be implemented for less than \$1000 using off-the-shelf hardware and does not require specialized knowledge or extensive consulting. Such disk arrays were scarcely available to consumers and small businesses even 5 years ago and, when available, required extensive consulting with storage experts, specialized hardware implementations, and cost tens of thousands of dollars or more. In short, disk-based storage is getting cheaper, easier, and more commonplace.

Disk operation is not all paradise, though. There are many issues to consider when operating disks. None of them should prevent you from using disk storage. However, they should be taken into account when implementing and operating any disk-reliant system. These issues can include:

- Disk lifetime—How long will each disk drive work before it fails?
- Throughput—How quickly is data getting from the storage system to the computer?
- Redundancy—Is the system truly redundant and fault tolerant?
- Fragmentation—Is the disk system operating at optimum efficiency?

Then in Chapter 2 we explored one problem, disk fragmentation, in great detail. In a nutshell, bits of a file get scattered all over a disk. This makes reading the file more difficult for the hardware to accomplish, decreasing the efficiency of disks and slowing down disk throughput. When critical files or often used files become fragmented the impact can be profound.

Fragmentation is a problem that can actually result from a number of different causes. Unfortunately these causes include normal daily operation of a disk. Over time, disks will become fragmented. There are preventative measures that we can take, and many that are designed right into our modern operating and file systems. But these measures only delay the inevitable.

The problems caused by fragmentation can generally be broken up into three categories: performance, backup and restore, and stability.

Performance

Most modern computer systems identify the disk storage system as a performance bottleneck. The CPU, memory, and data bus speeds increase almost as fast as customers can adopt them. But disk speeds have traditionally been slower to increase. This is mostly due to the nature of disk storage construction being based on moving parts including the rotating spindle and the read-write heads.

Disk throughput is serialized. The data must be transmitted to or received from the computer in a very specific order, because that's the order that the data makes sense. This can cause some performance issues when the disk cannot accept all of the data at once. The computer must wait for the disk to write the first part of the data before sending the next part. When the system must wait for the disk to become free for either read or write operations, the system's performance is often noticeably affected. This is illustrated in the following figure. Note how the file must be broken up. Each data segment requires its own operation, which naturally slows it down.

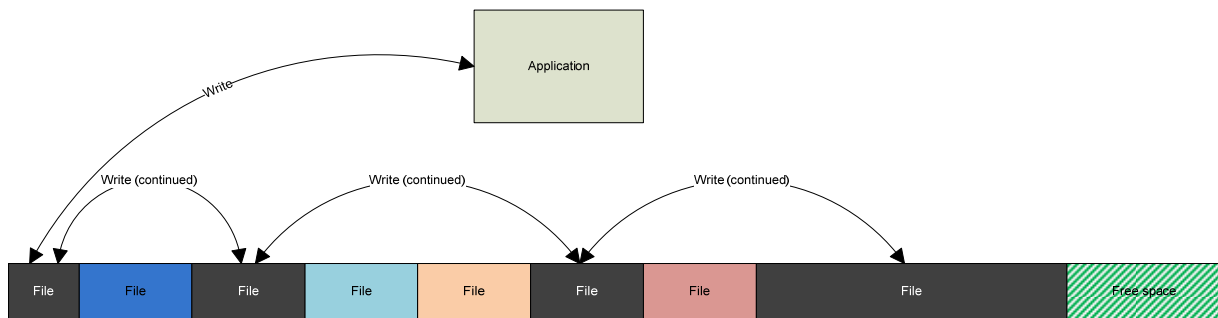


Figure 3.1: Writing a fragmented file. This file requires a minimum of four write operations just for the data.

When either reading or writing to disk, efficiency is critical to a quick and effective operation. Ideally all of the desired data is in one spot when reading from a disk, and a large enough piece of free disk space is available when writing to disk that the entire file can fit in one spot. Files where all the data is located in one spot are called *contiguous*. Contiguous files are the fastest to read and write because they optimize the disk's efficiency. Files where the data is in more than one location on the disk are *discontiguous* or *fragmented*. Depending on the level of fragmentation, these files can be very inefficient to operate on. The result can be read or write delays. The delays increase as the amount of fragmentation increases.

Backup and Restore

Companies that place value on their data go to great lengths to protect it against loss or compromise. They cannot lose the central focus of their business when a hard disk fails or a power supply blows up. These companies invest in a variety of data protection methods to minimize loss and downtime. One of the most basic and most effective methods is data backup.

In short, data backup is the process of copying data from a main source to a secondary source; for example, burning a DVD with a copy of a customer database stored on a server's hard drive. Normally, the data backup is carefully stored in a secure location so that when the original source is compromised, it is likely that the backup will be unaffected.

Data backup is often a slow process. There are several factors that contribute to data backup being slow:

- Volume of data can be enormous
- Inability to take the data offline, requiring a complex backup scheme to copy the data while it's being changed
- Scarce system resources available on data host
- Fragmented state of data source

Most standard data backup practices have built-in mitigations to these factors. They include scheduling backups during periods of system inactivity, purging unwanted data before the backup begins, and (less frequently) scheduling system downtime to coincide with data backup. However, many organizations ignore data fragmentation as a component of data backup. It's simply not part of their thought process. This is an incorrect oversight.

Data fragmentation can significantly impact a backup process. As we've already seen, fragmentation leads to delays in reading data from the hard disk. Data backups rely on reading data as quickly as possible for two reasons: to speed the backup process and to efficiently supply data to continuous-write devices such as DVD drives. Heavily fragmented data will take longer to read from the disk. Thus, at best, the backup process takes longer to complete. The worst case is that the backup will fail due to the delay in supplying data to the continuous-write data backup device.

Similarly, data restore is the opposite of data backup. It is the process used to take the information from a data backup and place it back in a usable state. This is most often a result of the primary data source becoming damaged or failing. For example, when a disk drive fails, the backup of the data from that drive is restored to another drive for continued use.

Usually data that is stored on a backup device is restored through normal file write operations. These operations rely on fragmentation avoidance techniques built in to the operating and file systems. However, the state of the disk at the time of restore plays a significant role on how the files are written. If the disk is crowded and there are few contiguous free spaces to write new data, the files will most likely be fragmented as they are written. This fragmentation will continue as the available free space becomes scarcer and even more fragmented. This issue is illustrated in the following diagram.

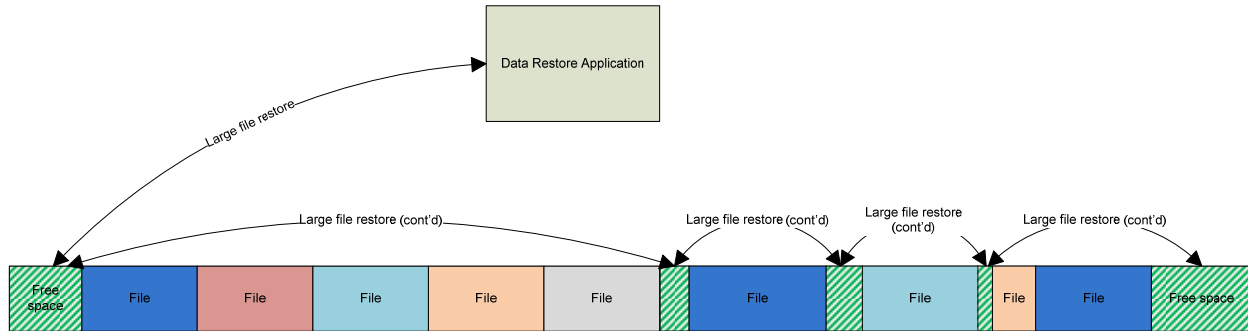


Figure 3.2: Restoring data to a disk with little or fragmented free space results in a fragmented file.

Stability

It is even possible that a fragmented system can go beyond slowdown to failure. Although rare, it is a possibility. The reason is that during OS boot, key system files must be loaded in a timely manner. The most likely cause for this scenario can be traced back to a heavily fragmented master file table (MFT) on a Windows computer running the NTFS file system. The MFT holds key information about all other files on the hard disk. Fragmentation of this file has a cascade effect, causing all other disk input/output to slow down while the MFT is read. Windows attempts to keep the MFT in a single extent but often fails to do so, especially when using a small or nearly full hard disk. Although other key Windows files can cause boot failures if they're fragmented, the MFT usually has the biggest impact.

If the key OS files are not loaded within a predetermined time limit, the OS may think that it has become compromised or corrupted. In that case, it will display an error message and stop the boot sequence in order to protect system integrity. This results in system downtime and could potentially require reinstallation of the OS to fix (unless you have a solution in place to defragment offline systems).

Addressing the Disk Fragmentation Problem

Now that we clearly understand how disk fragmentation works and why it is a problem, we can work on addressing it. Today, the most effective way to resolve fragmentation issues is to use a software-based disk defragmenter. These software packages can be highly effective and, over time, can not only eliminate fragmentation but also prevent it from reoccurring.

The remainder of this chapter serves as a guide to help you determine the best method to eliminate fragmentation in your environment. It is broken up into two sections:


- **Evaluating a Defragmentation Solution.** This section examines the various features of a defragmentation solution that should be considered when deciding which solution to choose. The most important areas are called out and decision criteria are provided to help you make an informed choice. No preference is given to any specific defragmentation solution. The decision is yours. This section merely helps you tell the different in features to help you make that decision.
- **Defragmentation Approaches.** There are two general approaches to running defragmentation software, automatic and manual. Each has benefits and drawbacks. This section calls out those benefits and drawbacks so you can decide which is best for your environment.

Evaluating a Defragmentation Solution

We understand that solving the fragmentation problem requires some type of software solution. But which one should we use? Luckily this decision isn't as hard as it seems.

The defragmentation software market isn't nearly as scattered or difficult to navigate as, say, the email server market. There are a couple of major players in the defragmentation market, several smaller niche players, and a solution built into Windows itself. Each of these solutions, even the most basic, has some benefits and drawbacks when compared to the other available options.

Our approach in this section is to examine the most common decision making criteria when evaluating a software product for wide scale deployment. For deployments of just five or ten computers, this type of exhaustive research and decision process may be overkill. It would probably more cost effective to just select a well-known software package and go with it. But if you've got hundreds or thousands of computers that need a defragmentation solution, it is best to perform a careful analysis before purchasing or deploying anything. This will help ensure that the software meets your expectations and solves the right problem in the right way.

 For more information on general software evaluation processes, consider reading about the [Software Selection Process](#). This process is presented by Technology Evaluation Center (<http://www.technologyevaluation.com/>) and is applicable to virtually any software evaluation and purchase decision. This web site has a number of tools and techniques that you can use to help simplify and improve your software selection process.

Cost

When we get right down to it, cost is always a consideration. Software licensing is never inexpensive, no matter what the software is. Some companies take “liberties” with their software licensing. The purpose of this paper isn’t to explore what a valid license agreement or use is. You need to determine that for yourself. But for the sake of this paper let’s assume that you’re going to purchase a license for each installation of the defragmentation software.

Most companies require a [cost-benefit analysis](#) before completing any type of large scale IT purchase. In the case of defragmentation software it should be no different. Throughout this paper we’ve examined how fragmentation can negatively impact an organization. The benefits to deploying the defragmentation software include increased productivity from higher performing computer systems without the need for hardware upgrades, stability of systems, and a higher degree of data integrity. All of these have direct value to an organization, and in most cases outweigh the cost of purchasing licenses.

When planning to purchase software, you should also take into account the indirect costs such as the costs of testing, deploying, and supporting the software. Although this is not a direct and specific dollar amount, it can be considerable. For example, you might evaluate two defragmentation solutions. One costs \$35 per installation, the other \$60 per installation. They both allow automatic background defragmentation and your research shows that they produce similar on-disk results. However, the \$35 solution requires interactive installation and must be manually updated when a newer version is available. The \$60 version provides extensive deployment automation and maintains itself over time. So which one is a better investment? The answer is that it depends on your environment and needs. Both are viable options. But you might not fully appreciate the subtle long-term differences between the options until you fully research and test both.

☞ Check both the initial and long-term costs, because you’re really investing in a solution for today and the future.

Defragmentation Engine Operation

When it comes right down to it, the defragmentation software needs to find all files that are fragmented and defragment them. This includes the operating system files, file system metadata, and all data files. Few files, if any, should be excluded, though some are harder to defragment because of how the operating system works or because of the limited impact of fragmentation on them. For example, the Windows pagefile is difficult to defragment because Windows itself puts a lock on the file whenever the operating system is running. So a different approach is required, such as defragmenting the file during boot time before Windows puts a lock on it.

Interestingly, virtually all commercial defragmentation software does this work. The results are about the same. Small features here and there might be different. Some defragment files slightly more efficiently or faster than their peers. Others claim to defragment in such a way as to leave bigger chunks of free space for future files to avoid fragmentation. But ultimately when all of the packages say that they will defragment the file system, in most cases they do it with roughly equal results.

How the engines get the files to a defragmented state, however, is an interesting area to explore. Consider that in Chapter 2 we learned that to defragment a file the software will read the entire file, locate contiguous free space where the file will fit, write the file there, and then delete the original file. This takes system resources. Disk usage obviously goes up when this is occurring. But CPU and memory are consumed to some extent as well. While the disk throughput can be slightly changed based on how the engine works, there's not a lot of wiggle room there. The significant difference between engines can come in CPU and memory utilization.

Obviously we want less CPU and memory used while the system is in use. For systems running 24 hours a day with consistent load, the only solution is really to simply ensure that the engine operates slowly and in the background, never interfering with the system's intended use. For computers such as desktops or workgroup-based servers, the engine should behave very differently. If all your users work from 9AM to 6PM, the engine should either be configurable to not use any resources that the system or other applications need within that time block or do so automatically. Once the users go home, it doesn't matter if the engine consumes 100% of system resources because the user will not be impacted as long as the process is completed by the time the user returns. You should look for software that allows you this flexibility.

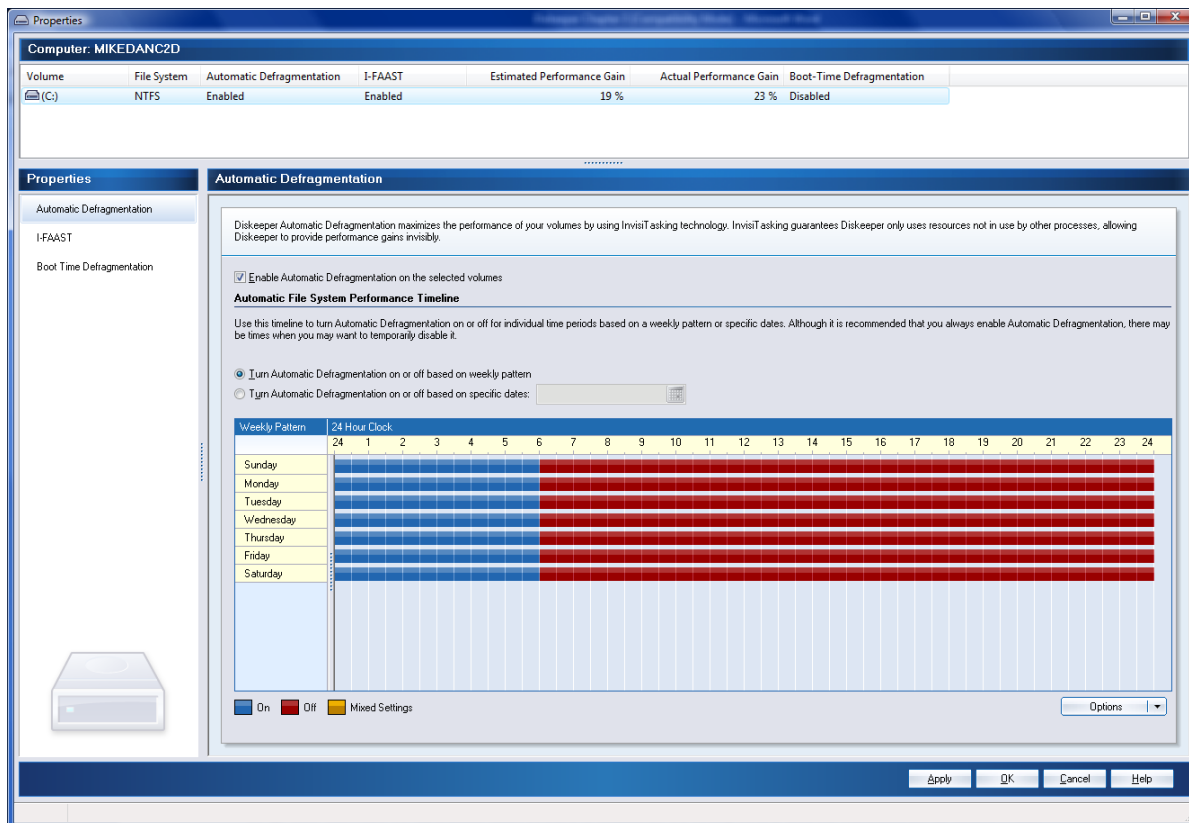


Figure 3.3: An example of how one software defragmentation package allows the user to configure what times defragmentation will and will not run. Note that in this case defragmentation is not forced to run, but is permitted to run if the volume requires it.

Most of the defragmentation engines produce the same result in many cases. However, the way they get there can be different. Ensure the software doesn't interfere with the operation of the system.

One feature that most high end defragmentation engines tout is the ability to prevent future fragmentation by optimizing the disk layout. The optimization method varies by software, and is often unique to that package. Although this can be useful for systems with high file throughput (e.g. file servers or systems that create a number of temporary files), its benefit could be somewhat limited. If the entire file system is frequently defragmented before fragmentation ever affects the user, it can be a far more efficient solution for optimized performance. In addition, the result of an optimized layout is somewhat theoretical because in a high-throughput scenario, and how the particular file system allocates the writing of new data, the operating and file systems will quickly refragment the system regardless of the free space layout.

Deployment

You'll need to get the software out to the client and server computers in your enterprise. How do you plan to do that? Walking up to each computer and installing from CD or USB memory doesn't scale past a small workgroup because of the labor and the potential inconsistencies when deploying in such a one-off manner. You must automate the deployment of your disk defragmentation solution to have any hope of a successful deployment.

Some software packages lend themselves to simple automated deployments by coming prepackaged and ready for distribution through mechanisms such as Microsoft Systems Management Server or Windows Group Policy. Usually this software is delivered in the form of a single Microsoft Installer (MSI) file. An administrator can simply take the MSI file, point their desired deployment software at it, and tell the software where to deploy it. Most deployment solutions are also good enough to provide scheduling and status updates to help the administrator know exactly how the deployment is going and whether there are any problems.

Digitally Signed Software

Digital signatures are becoming more common for Windows based software. The signatures are usually found on the installed software, but the need for digitally signed installer packages is quickly becoming important. Windows, by default, resists installing software that's not digitally signed because malware is very likely to be unsigned. To help ensure the smoothest installation experience and help support your company's security policy, you should seek out software solutions that provide both a digitally signed application and a signed installation package.

An automated deployment is the simplest and easiest way to get the software out to the clients. It is also the most consistent. Software-based deployments won't forget a desk because it's in the corner or miss Larry's computer because he has it in the back corner. There are a number of other reasons to recommend this method of deployment. The time and money savings, combined with the reasonable consistency and completeness of an automated deployment, makes it the preferred method.

Undeployment

We understand the need for some method of automated deployment. But what happens if the software doesn't produce its intended result or doesn't get funded long term? Can you take the software back from the systems you deployed it to? This can become important, especially in cases of licensing where you no longer have permission to use the software. Failure to undeploy the software consistently may result in liability or a destabilized environment. You should ensure that the software will come out just as easily as it comes in. This can be proven during the testing process, described later in this document.

The only exception to an automated deployment should be disconnected or remote computers. Usually this means laptops and edge servers. These computers usually are not joined to the domain and frequently have no automated maintenance methods available. However, they are just as in need of the defragmentation solution as the rest of the computers, if not more so. In these cases, consider using a network map or list of computer assets to ensure that all computers receive the proper software installation.

☞ Deploy your defragmentation solution through automated software installation wherever possible. Examples include Microsoft's System Center Configuration Manager (formerly SMS) and Windows Group Policy. For computers that can't receive automated software delivery, do it the old fashioned way – by hand.

Operational Autonomy

Once the software is deployed, it begins to operate. How well does it do on its own? Most software is good about using default values to begin operating properly. Even if the software cannot get additional setting changes from a central location like Microsoft's Group Policy or a software-specific administration server, it should be intelligent enough to begin operating without any additional information.

Over time the use of a system may change. Perhaps a disk is added to the system. Does the software recognize that and adapt itself? The more effective software solutions do. If specific setting changes are required whenever a system configuration is modified, the cost of administering that software solution goes way up. This should be something you consider long term.

☞ If the software can operate itself with little or no administrative interaction, that's usually a good thing. Self updating and configuring with intelligent defaults are the two biggest features in this category.

User Experience

As we found earlier, for the most part the defragmentation engines do the same thing. They reassemble bits of fragmented files into one big unfragmented file. Users care about this indirectly because they want higher performing, more reliable computers. But what users and administrators truly care about is the simplicity of using the software. They care about the user experience.

Defragmentation software has made tremendous inroads in the user experience area over the last decade. Some of the earliest solutions were nothing more than a command-line entry of

```
C:\> Defrag.exe
```

This resulted in a prompt telling the user to wait until defragmentation was complete. This is an effective if unpleasant and cold experience. Through the years the experience changed, most notably with the disk space “kaleidoscope” where data and files were represented by different colors – red for fragmented and green for contiguous files, for example.

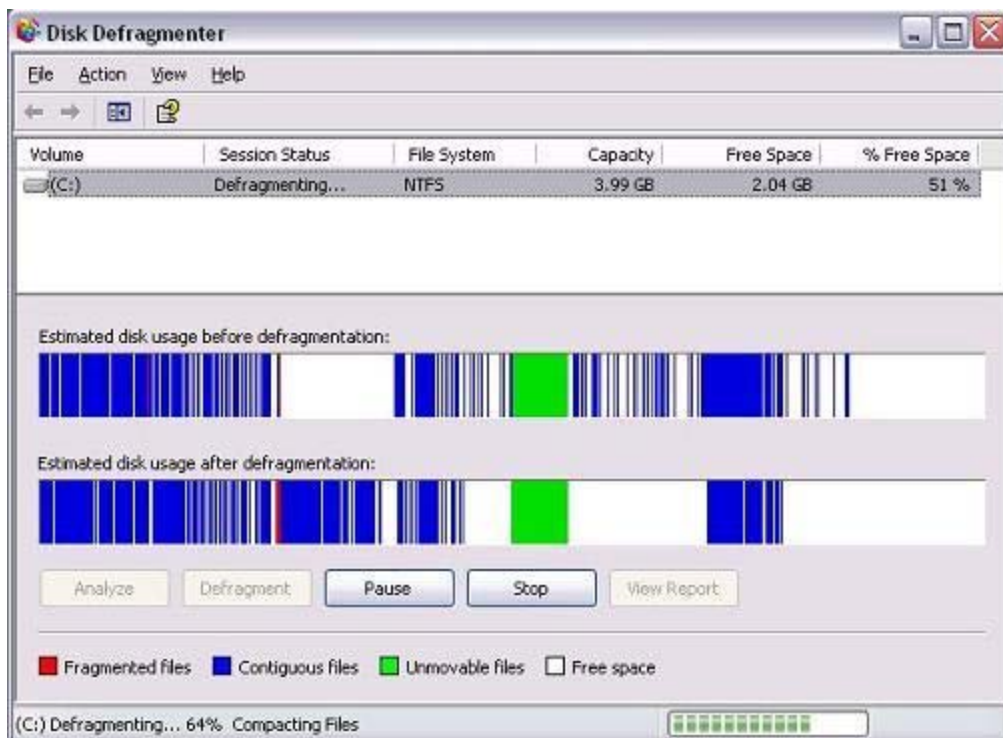


Figure 3.4: Windows XP's built in Disk Defragmenter tool with its basic “kaleidoscope” display.

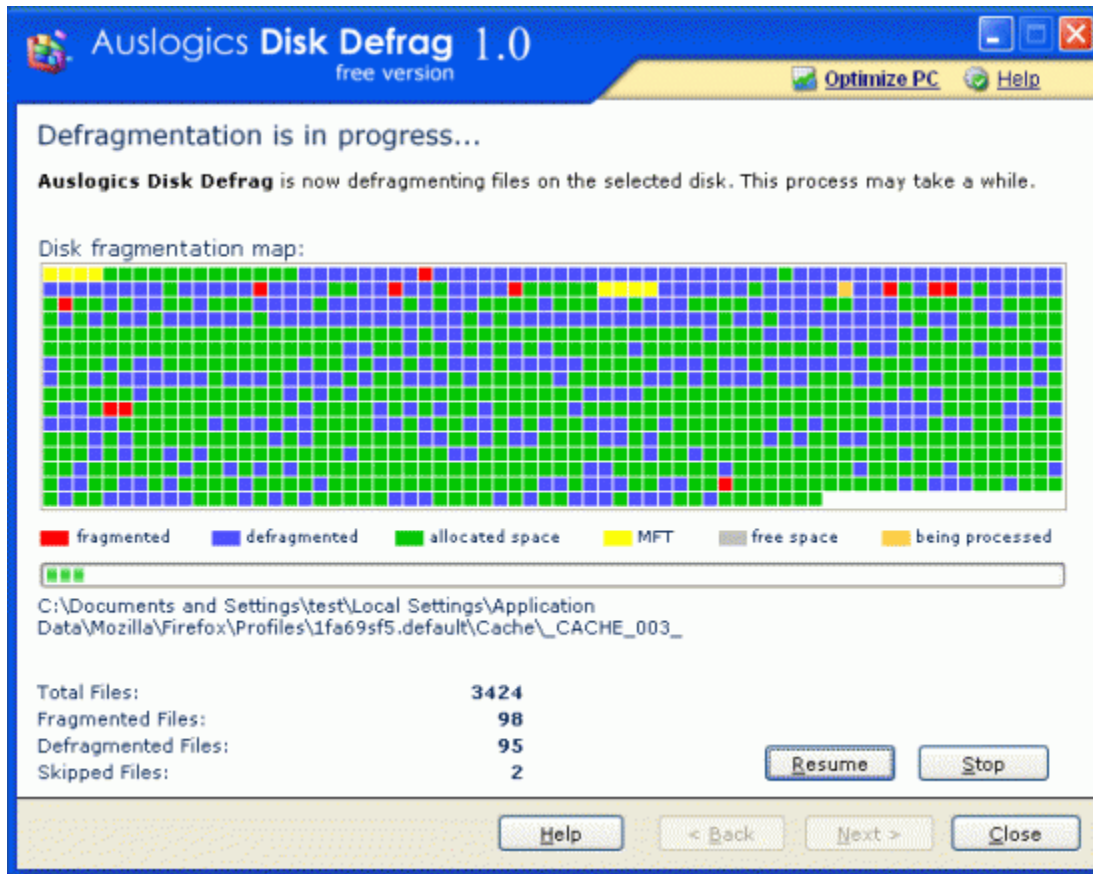


Figure 3.5: A typical defragmentation software package. Note that the display is almost identical to the Windows XP built in utility except the colors are small boxes instead of tall lines.

Today, the user experience for most defragmentation software is an effective balance between pleasant graphics and task-oriented objects to help the user understand status and make decisions. For example, this is what a typical screen looks like from the Diskeeper defragmentation solution:

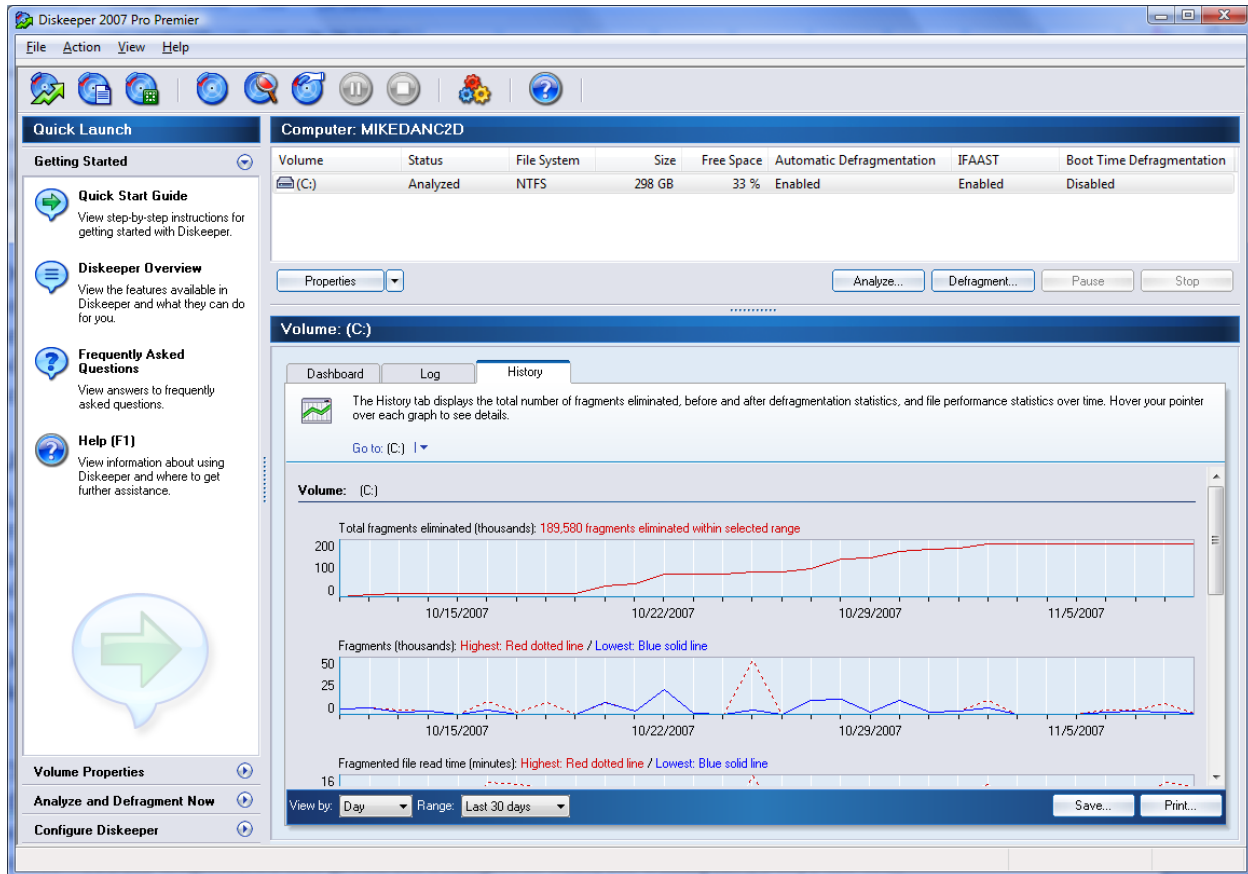


Figure 3.6: Diskeeper 2007 Pro Premier. Far different than the command-line defragmentation software of a decade ago.

Usability isn't everything though. A pretty shell with no substance behind it isn't very effective at improving system uptime or performance. Luckily most defragmentation software concentrates on the engine before polishing the shell. But when in doubt, the usability should be considered less important than the operation of the system itself.

☞ The user might never see the defragmentation software depending on how you deploy it and how it operates. So the usability feature might be chiefly for administrators. Ensure that the person that needs to operate the software understands it.

Reporting

Once we've decided on our preferred defragmentation solution, tested it, and deployed it, are we done? For some administrators, the answer is yes. But for most of us, the answer is a resounding "No!" We need to ensure the solution is working properly, both immediately after and long-term as a sustained operation. This is where reporting comes in.


There are two main functions for reporting on disk defragmentation solutions. One is to verify that the software is, in fact, installed and functioning properly on the desired computers. Getting a daily report that shows that the defragmentation software operated properly is very useful. If the computer reports an error or fails to report, an administrator can address it before the computer is impacted by excessive fragmentation.

The other reason we gather information on defragmentation results is to see whether the software is making a difference. Whenever we spend money on software solutions, we want to be able to measure the impact that this software had. Simply telling the CIO that "The computer was defragmented and that's a good thing" doesn't really support an ongoing business model. However, telling her that the defragmentation solution has removed over 19,000 fragments per week and that resulted in a 7% increase in throughput on a central server is a very significant statement and can easily justify the software investment that you've made.

If your organization wants to provide ongoing justification for your defragmentation investment, or you want to ensure that the software is installed and operating properly over time, consider obtaining a software package that provides rich reporting features. Some packages simply put an entry in the Windows Event Log that contains little more information than "Defragmentation job ran." While this might be enough for you, other packages contain much more information. Some data you might want to gather, depending on your specific organizational needs, could include:

- Defragmentation start and stop time
- Number of file fragments defragmented
- Condition of file system (e.g. NTFS metadata)
- Version of software running

All of this information should, optimally, be compiled into a report. Although there are any number of great reporting software packages available, the better defragmentation packages already have most of that functionality built right in. They can provide detailed analysis about performance impact and, from that information, you can clearly show the benefit and justify the cost of the investment.

 Reporting is key for initial installation verification and for ongoing maintenance. If you plan to track the defragmentation software over time, ensure that the software has the capability to report on itself. Also check to see if it requires other software to coalesce reports, as this might be an expensive prerequisite.

Defragmentation Approaches

Once you've determined the best defragmentation software solution for your environment, you're ready to decide on an approach. There are two categories of defragmentation approach, manual and automatic. We'll explore both of these in this document.

You can actually explore both approaches and test them in your pilot or test environments to help you make a decision. Although it's easy to decide on one approach or the other, most of the time you'll need to have one as a default approach and make exceptions where the other is most appropriate or the only viable option.

Automatic Defragmentation

As its name implies, automatic defragmentation just happens. On a regular basis (usually in the middle of the night), the defragmentation software wakes up, scans the drive, and if necessary performs a defragmentation and clean-up operation. Usually the system reports the results of the operation to a central server (see Reporting in this document).

This approach seems to be the obvious choice. And for most environments and applications it is the best way. The benefits to using automatic defragmentation include:

- No user or administrator interaction required, making the experience easier and more likely to succeed
- Predictable operation
- Customized run time to ensure little interference with normal system operation

Almost all defragmentation solutions default to some type of automatic configuration. A few older or more limited solutions require you to configure them as batch jobs or via scripts to run automatically. These are not recommended because they're more likely to fail and will almost always lack any reporting features. And although this isn't a direct correlation, most applications that have to be executed in this way consume enormous system resources as they believe that the user is interactively executing the defragmentation process. This could result in resource issues if the process is triggered at the wrong time or runs too long.

There may be situations where you do not want to use automatic defragmentation. You might want to completely control when and how the defragmentation software operates. In those cases, you'll need to use the manual defragmentation approach.

Manual Defragmentation

Automatic defragmentation works in the majority of situations. The more automated the solution, the less we have to worry about mistakes or errors causing a breakdown in the system. And for most systems, regular and predictable defragmentation is the preferred solution. But, depending on the product, there may be times where an automatic solution is just not going to work. Let's look at an example.

You are in charge of a small web server cluster that consists of two web servers and one database server. This cluster has very specific performance requirements that it is just barely meeting. You run backups and maintenance whenever the load is low. Unfortunately, due to the nature of the traffic and the users that hit this cluster, you never know when a spike or sag in traffic will occur. In this case, an automatic solution might kick off a resource-intensive defragmentation pass in the middle of a usage spike. That could easily push the performance outside the minimum requirements. If this is a possibility with software, the better method would be to manually kick off the defragmentation pass during a lull and, if traffic picks up before the conclusion of the pass, stop the defragmentation and resume it later.

Some defragmentation solutions include logic that will automatically perform defragmentation when the system reaches an idle state, and stop the process when the system begins being used. This works well for some applications such as desktop and laptop computers, and depending on the technology, for servers as well.

How to Make Your Decision

You've looked at the available defragmentation solutions. You've decided on a default defragmentation method and potential exceptions. You have an idea of how many computers will receive the software and how it will be deployed. Now you need to make your purchase.

The remaining phases of the purchasing process are fairly straightforward. These are common to any software evaluation decision and include:

- Preselection
- Test
- Purchase
- Deployment

Let's take a brief look at each of these phases.



One phase of the process not mentioned here is operations, which is sustaining the software in the environment over time. Because the sustained operation of defragmentation software solutions is so similar to other software solutions, it is not included here.

Preselection

Now that you've identified the needs of your organization, take a look at the solutions available. There are a number of ways that you can find out information about the features of the software packages. These include:

- Download a demonstration or limited copy of the software
- Read whitepapers from the software developer
- Check software reviews from other corporate users
- Visit the company's web site
- Ask the manufacturer to have a sales representative contact you
- Network to find others who use the same software and ask them their opinions

The desired result of this work is that you'll have one or two solutions that you believe will work best for your needs. There may be a long list of potential solutions, but using these methods against the decision criteria we developed earlier should help bubble the best candidate to the top. Once that happens, we can examine the best candidate through testing.

Test

If you asked a hundred of your peers whether they would test a preselected software package before deploying it in their environment, I'm sure 99 of them would say yes. In fact, you're probably wondering why this section is more than one simple sentence, "Test before you buy." The reality of the situation is that, while the statement is true, there is a bit more to it.

There are a number of things you should look at during your test suite. Hitting these will help ensure that you make the right purchase decision and that future issues with the software are minimized and well understood.

Some basic test methodology and items to look for during the process include:

- Setup an isolated test environment to minimize impact on production resources
- Ensure the test environment is representative of the entire production environment
- Ensure the test deployment mirrors the intended production deployment
- Test normal use cases, such as a user running Microsoft Word while the system defragments a minimally used drive
- Test edge cases, such as a system under 100% load while the defragmentation process engages on a near capacity and heavily fragmented volume
- Verify that the reporting component provides desired data
- Verify that the engine updates itself if necessary
- Document your findings
- Consult the software manufacturer for assistance with unexpected or undesired results
- Ensure the software can be undeployed gracefully

This may sound like a lot. But with virtualization and a small hardware investment this can be accomplished on one or two computers with just a few days of work. Once this set of tasks is complete and you are satisfied with the results, consider performing a pilot deployment. Choose a small number of users and computers that are representative of your organization and deploy the software to only them. You can then measure the impact in production without the potential for widespread impact to the organization.

Deployment Guide as a Result of Test

Most organizations overlook one key element of testing that often justifies the entire process. During test, you have to deploy and redeploy a number of times. And you're documenting the process as you go. A natural result of this work should be a Deployment Guide for the software that you can use in production. This detailed guide will be fully tested and verified before the end of the test process. It is an invaluable document for your deployment staff because they can understand exactly what steps to perform, what results to expect, and how to handle any variances that may occur. And if you're doing a thorough job of testing, this document should require virtually no additional effort.

Once you've completed both your isolated test suite and your pilot deployment, you should have enough information to decide whether to proceed with the purchase and widespread deployment of the software. But do not be surprised at this point if the project takes a different direction. The results of applied testing sometimes help us draw different conclusions than we had previously thought. For example, you might find that your preselected and tested defragmentation solution conflicts with a disk management application that you use on 25% of your computers. In that case, you would be unable to proceed with the deployment, at least to those affected computers. You might decide not to deploy any defragmentation solution, to those computers, to use two defragmentation solutions, or to test your second choice to see if it also has the same issue. But obviously it's better to find this out before you've purchased licenses and begun your widespread deployment.

Purchase

The purchase process will be different for every customer and every software vendor. Virtually every purchase is going to vary to some degree. So providing specific details here isn't really useful. Some software vendors will negotiate bulk pricing, while others will not. Some will accept purchase orders or incremental purchases at the same discount, others will not. You might receive your software funding over time or all at once. The possibilities are endless.

The one element that you should concern yourself about is what you're buying. Is the software copy protected? If so, will you receive one license key for your entire purchase, or a different key for every seat? This can dramatically affect your deployment, so make sure to check with the software vendor. You do not want to have to walk to each computer and type in a unique 35 alphanumeric character key! That would be more painful than a visit to the dentist's chair and far less productive.

You should also ensure that you receive some number of retail shrink-wrapped packages. These are effective as known good, clean copies for building system images and performing test installations from local media instead of the network. Most vendors are happy to supply a handful of these, which should be kept under lock and key per your company's software retention policy. You should ensure that you have enough on hand in case of problems like the loss of a software deployment server or having to install the software to an isolated or offline system. Some software companies offer alternative methods for software acquisition and storage, such as online libraries or the option to burn their software to CD/DVD on demand. Use whatever method you're comfortable with, as long as you have access to a backup of the software in case of emergency.

Deployment

Great! You've analyzed the market, selected a software package, tested it thoroughly enough to know it works for you, and purchased enough licenses to begin your deployment. Now let's get going!

The section on deployment earlier in this document covered the majority of deployment considerations and decision criteria that you make. But by the time you get to this stage you almost certainly have a very specific deployment strategy, plan, and documentation. Now it is time to execute on your well thought out and documented strategy.

In a perfect world, the deployment is the easiest part of the process. But in reality, issues will arise. Conflicts will come to the surface that weren't detected during testing. Your deployment software might hiccup and miss a hundred users. The new software might conflict with another application that's only deployed on a small number of computers so you missed it during testing. Regardless of how well you planned, remain flexible and deal with the snags as they arise.

Consider Standardizing and Automating Your Deployments

If your organization does not currently have a standardized deployment strategy, you should consider investigating this option. The benefits to having one are almost too numerous to list but include ensuring IT consistency across the organization, more effectively managing software licenses, and reducing the total cost of ownership of systems by reducing the deployment time of a computer from hours or days down to minutes with little or no administrator interaction. Consider reviewing Microsoft's [Business Desktop Deployment 2007](#), which includes both guidance and automated tools and is available for free download.

At the end of the deployment phase you have your solution installed and running on all of the intended computers with the software verified and reporting its status. But deployment isn't really ever complete. New computers come into the environment and require one-off deployments. Old computers require undeployment or reconfiguration. This is part of the ongoing software operation lifecycle, but it is the same as any other piece of software.

Summary

Selecting a defragmentation solution can be a difficult process. There are a number of factors to consider. Beyond the basic surface considerations such as cost and features, there are things such as operational autonomy and ease of deployment that will contribute to the long term cost of such a solution. Having a list of things to look at and a process to follow helps us along this process considerably.

In Chapter 4, we will examine the business side of defragmentation. We'll talk extensively about the return on investment strategies and justifications for using defragmentation in the enterprise. We'll talk about cost-benefit analysis in some depth. Chapter 4 is written primarily for those in a decision-making role for enterprises to help with the cost justification of the purchase.

Chapter 4: The Business Need for Defragmentation

Chapter 1 explored how disks work. They used to be large, expensive, slow, and have very limited capacity. Today, that has all changed. Modern disk storage is inexpensive and provides nearly infinite storage capacity for a small investment. Modern laptop and desktop computers routinely have a terabyte or more of storage, a capacity that was unheard of on even the largest systems 10 years ago. Such capacity comes with a reasonable price tag and maintains very high performance if properly maintained.

Disk operation is not all paradise, though. There are many issues to consider when operating disks. None of them should prevent you from using disk storage. However, they should be taken into account when implementing and operating any disk-reliant system. These issues can include:

- Disk lifetime—How long will each disk drive work before it fails?
- Throughput—How quickly is data getting from the storage system to the computer?
- Redundancy—Is the system truly redundant and fault tolerant?
- Fragmentation—Is the disk system operating at optimum efficiency?

Then, in Chapter 2, we explored one problem, disk fragmentation, in great detail. In a nutshell, bits of a file get scattered all over a disk. This makes reading the file more difficult for the hardware to accomplish, decreasing the efficiency of disks and slowing disk throughput. When critical files or often-used files become fragmented, the impact can be profound.


Fragmentation is a problem that can actually result from a number of causes. Unfortunately, these causes include normal daily operation of a disk. Over time, disks will become fragmented. There are preventative measures that we can take, and many that are designed right into our modern operating and file systems. But these measures only delay the inevitable.

Chapter 3 divided the problems caused by fragmentation into three categories:

- Performance
- Backup and restore or data integrity
- Stability

Each of these is a key driver in the return on investment (ROI) for every computer system in an organization. Certainly a system's performance and stability directly affect the productivity of that system and anyone who relies on it. For example, if a user's desktop computer is unstable, the user's performance rapidly degrades. Extending that to a server, now the performance of every user who relies on that server is degraded. All these detrimental system complications quickly add up to significant loss, whether directly observed (for example, system downtime, data loss) or more subtle but just as real (for example, small degradation over time).


We then explored a decision-making process for selecting and implementing a defragmentation solution. This process was based largely on technical criteria, as the intended audience of Chapter 3 is the IT department, including the IT implementer and the IT manager. However, that process often includes, and in many cases is owned by, the IT department's business decision maker (BDM). The BDM needs a different set of criteria because their needs and responsibilities have a different focus from the implementer's. Those BDM decision-making needs are the subject of this chapter.

 You should review Chapter 3 thoroughly either before or after reading this chapter. They go hand in hand to provide a complete picture of the selection and implementation process. Although some of the content will be similar or duplicate, other content will be unique and prove very useful to understand the problem and solution from more than one viewpoint.

In this chapter, we will examine the fragmentation problem as a business problem. First, we'll spend some time looking at fragmentation as a business risk. Although previous sections described the on-disk technical details, we'll look at the impact to users, systems, and the business. Once we've seen what kind of impact fragmentation can have, we'll take a look at how best to justify a solution to the problem. The best way to do this is with case studies. We'll examine examples of other companies that have successfully mitigated the fragmentation problem and use that data to help justify our own solution. Then we'll provide a strategy for selecting a defragmentation solution. Previous chapters examined this same problem from a technical perspective, but we'll examine the problem from a business standpoint. For example, the technical solution may not account for an ROI calculation as part of the solution. However, from a business perspective, if the solution isn't worth more than the problem, we may not fix it at all.

Understanding the Investment

As we saw in Chapter 3, there are a number of problems caused by disk fragmentation. As previously mentioned, we divided the problems caused by fragmentation into three categories. Let's briefly recap these categories and explore how they apply to our ROI decision-making process.

 For more technical details about how fragmentation impacts these categories, see Chapter 3.

Performance

When a computer's disk is fragmented, more read-and-write operations are required to manipulate the same amount of data, and these operations become more complex as the data is further fragmented. Over time, most disks become fragmented. This means that over time, system performance can degrade as a result of slowly escalating fragmentation. Depending on the severity of the degradation and the amount of time that the symptoms took to manifest, it may take time before fragmentation is perceptible to a user. Let's look at each problem individually because, even though they usually have a common root cause, the approach and remediation to each can be very different. And so can the cost to repair each type of problem.

User-Perceived Performance Issues

IT and Help desk staff encounter user-perceived performance issues frequently under the generic complaint of, “My system used to be fast, but now it takes forever to do anything.” There are numerous potential causes of such problems, and disk fragmentation is one of them.

When the IT department receives performance complaints, they usually have a standard set of tasks and tools that they use to help improve the overall performance of the system. These often include actions such as:

- Rebooting the computer
- Emptying the Web browser cache
- Deleting files in the Temp directory
- Defragmenting the hard drive using the built-in Windows defragmentation utility
- Scanning for viruses and malware
- Running Windows Update to apply any outstanding patches
- Uninstalling all user-installed applications
- Re-imaging the system as a brute-force fix

Fortunately for many users, one or more of these steps usually results in some measurable system improvement. As a result, the user stops complaining and the IT group closes out the ticket. Although this might seem like a good thing, there are a number of flaws with this strategy. The primary flaw is the implementation of these steps. They are usually done as one combined suite of problem-solving steps. If the computer becomes acceptably performant (or is “fixed), there is no way to determine which step was responsible.

In addition, almost all these steps are one-off performance improvement tasks. None are effective in improving the system’s performance permanently or repeatedly except the Windows Update task, which is virtually never going to improve the system’s performance anyway.

The cost of responding to this complaint is significant. The loss of productivity is the most obvious concern, because the slower the user’s computer, the less efficient the user is at performing computer-intensive tasks. Between the time the performance loss is recognized and the problem is reported to IT, the user generally spends some time complaining about it to coworkers and management. Once reported, the IT staff might take several hours to run their suite of repair tasks and restore the system to a “usable” state. All these are significant money and time drains on your resources.

Less-Perceived Performance Issues

System slowdowns are often not very obvious to users. Consider a system that degrades its performance at a rate of 1% per hour compared to 1% per week. The former system would have a significantly slower response after just a few days. The latter might never be detected by a user until some time-critical task was performed or until they used another system that did not have the same performance issue. Many users never notice long-term system degradation at all or simply attribute it to a system getting old. They wrongly assume that computers, like people, get slower as they age. A computer system should perform equally on its first day and after a decade of use. Modern computers don't wear out or degrade like people or older machinery. They either work or they fail.

However, there is a twist. Computers perform properly only when maintained. There are only a few ongoing maintenance tasks that must be run at a regular basis, but they are critical to keeping a computer performing at peak levels. These regular maintenance tasks include:

- **Scanning for and removing malware.** This should be done on a daily basis and is usually also done in real-time by software that remains active on each computer. Although many enterprises maintain firewalls and security gateways, each computer is also usually configured with malware protection for situations where the malware avoids the perimeter defenses.
- **Defragmenting and preventing fragmentation of disks.** This is also done daily and can, depending on the software solution, be done continuously in real-time. This must be done on each computer throughout the enterprise, regardless of its role.

This guide concentrates on the second task, defragmentation. This can be done on a regular or continuous basis to help ensure that there is no performance impact to users. Almost all defragmentation software packages have the ability to run at off-hours times on a daily basis. The more advanced packages also contain technologies that both help prevent future fragmentation and defragment continuously in the background. These are very useful as fire-and-forget solutions so that the administrator and the user can feel confident that this problem is addressed with no interaction required.

Example of Fragmentation Performance Impact

Fragmentation is often cited as a detriment to system performance; however, surprisingly few hard facts have been published on the impact of disk fragmentation in a sizeable organization. This chapter will offer several examples of measurable impact through the citation of case studies and other published data. The first example is in the area of performance impact at a worldwide restaurant chain.

Consider a case study written by Joel Shore in 2007 entitled, "Diskeeper Keeps the Food Coming at Ruby Tuesday." In this case study, Shore explored the impact that disk fragmentation has in this worldwide organization of more than 900 restaurants. With such a disperse organization, a lack of centralized IT assets is almost guaranteed. In addition, the likelihood of advanced user knowledge at each restaurant, or even in each region, cannot be assumed. Thus, there is no IT staff to perform regular speed-up or system maintenance tasks. However, Ruby Tuesday identified defragmentation as a requirement for all their systems worldwide to help ensure ongoing system performance.

As a result of the lack of local IT staff, one of the key drivers to their solution selection was that they required a hands-off solution—that is, the solution just work with zero input or decision-making by the user. After they purchased and implemented their solution, Ruby Tuesday estimated that they potentially saved \$2.1 million per year by keeping the systems performing at peak level.

Data Integrity

Fragmentation does not just have an effect on the system's performance. There is also an impact on the integrity of the data that the system stores and processes. As a system's disk becomes fragmented (whether slowly over time or rapidly due to significant data throughput), the number of discrete read-and-write operations necessary to manage the data increases. Each operation is an opportunity for the disk to fail and puts a little more strain on the mechanics of the drive.

Consider your car. If you keep your car at peak performance, it is less likely to break down. Regularly changing the oil means that the engine encounters less friction and requires less effort to move the car. If the oil gets dirty and old, the engine has to work harder because there's resistance. Harder work means shorter life for engine components, and as a result, you're more likely to encounter a breakdown or mechanical failure.

Although disk drives aren't nearly as reliant on this type of upkeep, there is a measurable difference between a well-maintained drive and one that has had no maintenance. A drive that has severe fragmentation works very hard to read and write data compared with a similar drive without fragmentation. Less work means less likelihood of failure, which means increased data availability and integrity.

Unfortunately, data integrity issues usually do not provide advanced warning. They usually manifest when a user tries to access data and the file is missing or corrupt. At that point, the best alternative is usually restoring from a backup or looking for a copy (for example, a recent copy sent via email). Once this initial data integrity issue is recognized, most administrators will immediately take steps to verify the integrity of other data and proactively mitigate any other data integrity issues (for example, get a complete backup of the data, repair the drive, and so on).

Data Recovery Services

There are a number of data recovery companies in business today. These companies specialize in recovering data from failed computer hard drives. They often charge anywhere from a few hundred to a few thousand dollars for their service depending on the quantity of data, the age of the drive, the level of damage to the drive, and other factors.

Before you are in a situation where you need this kind of service, you should consider performing regular data backups. If you have a reliable backup copy of your data, you are less likely to need this unreliable and expensive service. This is especially true for irreplaceable data such as photographs and email conversations which may be difficult or impossible to recreate.

Stability

Within the context of fragmentation, data integrity and stability are very similar. If the system's data integrity begins to fail, the stability will also fail. This is because the computer's operating system (OS) is really just made up of data on the hard drive, just like any other data. If the data begins to become compromised, as discussed earlier, the system's stability will decrease.

The symptoms of an unstable Windows system vary widely but can include:

- Random system hanging or halting
- Periodic crash dump or "blue screen" error messages
- Irregular application error messages, often not corresponding to any specific action
- Noises from the hard drive (almost any noise from a hard drive is a sign of trouble)
- Poor system performance, often including moments where the system pauses for a moment
- Random system reboot or shutdown events

As you can see, some of these symptoms are very severe. In many cases, they can have a profound impact on the system's usability and on the user's confidence in the system. If you've ever lost several hours worth of work when a system unexpectedly crashed, you can relate to this problem.

Luckily, system instability stemming from fragmentation usually has some early indicators. Most systems don't just suddenly stop working as result of excess disk fragmentation. There will usually be one or more of the previously mentioned symptoms that worsen or become more frequent over time until they are addressed or the system completely fails. This can allow an IT professional to step in early and mitigate the problem before it becomes catastrophic (for example, making a data backup, defragmenting the disk).

The early indicators are both a blessing and a curse. Regardless, the problem needs to be addressed. Luckily, you can make a smart investment to fix this problem before it manifests itself. The next section discusses exactly how you can justify the investment and ensure that your systems remain reliable and your data remains intact.

Justifying the Investment

At this point in our series on disk fragmentation issues, you should understand many of the problems that fragmentation creates or exacerbates. You are probably looking to implement a solution immediately. Chapter 3 covered the technical aspects of implementing a solution. But as a BDM, you need different data. You need to understand the ROI to justify the spending to stockholders, management, ownership, and so on. In many organizations, you are required to write a formal justification for spending this amount of money. You may also need to create a change control justification for your IT department before they will deploy a change across all computers. This section helps you create this type of content.

We will examine the justification for our purchase in the same three categories that we've been using to describe the problem: performance, data integrity, and stability. For each section, we'll look at case studies of companies that have realized quantifiable improvements in these areas. These case studies often overlap, providing improvements in two or all three categories. However, there is usually one area that stands out more than the others.

A Note on Case Studies

This section uses case studies to justify a business decision. In most instances, these case studies have been commissioned by companies that distribute or sell defragmentation solutions such as Diskeeper Corporation. Regardless of the source of the research or the funding behind it, this document will continue to examine the fragmentation problem and solutions without bias to any particular vendor or solution.

Performance

As we saw earlier in this chapter, higher performance often directly leads to more effective workers and faster processes. When computer performance is at its peak, we realize efficiencies across a variety of assets. People get their work done faster (and spend less time complaining about slow computers), process-intensive, or throughput-intensive tasks run faster, system backups run faster, and so on.

Improving Employee Productivity

We already understand that increased employee productivity is a benefit to any organization. There are many examples of improving employee productivity through disk defragmentation. One great example that was previously mentioned is the restaurant chain Ruby Tuesday. This chain has restaurants around the world, which presents IT challenges, as there is rarely a technician at the restaurant. The computers at each restaurant must be self-sufficient and require little external maintenance over time.

As we learned earlier, all computer systems encounter disk fragmentation over time as part of the normal operation of the system. Ruby Tuesday identified that the ongoing fragmentation of their systems was causing each computer to slowly lose performance, which resulted in their ordering and billing processes (the processes that required the computers) to slow. Because the success and profitability of restaurants often depend largely on their efficiency, Ruby Tuesday focused on identifying the cause of the system slowdown.

In addition to the system's decreasing performance, the risk of having a hard drive fail prematurely is significant. Consider the difficulty and cost of replacing a hard drive in a remote restaurant where there is no IT presence and no trained personnel locally. Computers might be down for days or weeks, or entire computers might need to be delivered and replaced (again, with trained personnel). This results in a very high cost whenever a system fails.

Ruby Tuesday calculated that a loss of 30 seconds of productivity per hour due to computer performance issues in a restaurant open 12-hours a day resulted in an annual potential revenue loss of \$2.1 million (May 2007). That is a significant loss of revenue in any organization, and in a restaurant chain where competition is high and profit margins are thin, this can make an enormous difference in the company's success.

One choice that Ruby Tuesday made was to implement a hands-off disk defragmentation solution. The solution they chose ran automatically with no user input. Although the deployment, results and configuration could be centrally managed by the centralized IT department, the daily operations were completely automated. This kept the long-term operating costs low and ensured that local staff at each restaurant did not need to be trained in IT operations (another costly investment considering the employee attrition at most chain restaurants).

Increasing System Longevity

Another example where we realize a monetary gain from higher performing computers is in capital expenditure and asset longevity. A very common reason for replacing computers in an enterprise is in response to users complaining about insufficient system performance. Many organizations have a well-defined longevity requirement for computers, but the user complaints often drive review of, or exceptions to, this process. But the longer a computer is used, the higher the return on that investment becomes. Therefore, we want to use the computer for as long as we possibly can.

Defragmentation helps in this area by improving performance and therefore increasing system longevity. As you saw in previous chapters of this series, the performance between fragmented and defragmented systems can be dramatic. Even a minor change in the usable life of a computer system can be dramatic when you consider several factors:

- The number of similar computers that need replacing
- The hardware cost of replacement
- The operating cost of installing, configuring, and transferring data to the new systems
- The disposal cost of the existing computers

All these factors help us understand that obtaining even a small extension in a computer's usable lifetime is a significant cost-saving measure.

Decreasing Deployment Time

Another area where performance plays an important role is in the deployment of new computer systems. Consider how your organization sets up and configures new and replacement computers. Most likely you use system imaging software that installs images over the network, such as Symantec's Ghost or the built-in imaging software installed with Microsoft Windows Server 2003 and 2008. These processes are dependent on both the disk and network to efficiently transfer an enormous amount of data to the new system. Any improvement to disk throughput will improve the efficiency of the network-based data transfer and therefore improve the system's imaging speed.

One example where defragmentation made a difference in system imaging speed was at the Trinity School in New York City. The school's director of technology, David Alfonso, used a defragmentation solution throughout his data center. One significant improvement he realized was in system imaging, where he saw the time required to load a system image decrease from 25 minutes to 12 minutes. The decrease in disk fragmentation improved local data throughput which, in turn, enabled the imaging software to more thoroughly use available network bandwidth to keep the data stream moving at the highest speed possible. Because the Trinity School images up to 20 systems at a time, the performance improvement helped Alfonso realize an enormous efficiency in this area.

Data Integrity

The cost of losing data can be significant. Consider a few scenarios:

- A bank relies on its database to log customers' transactions and account information. Even a single point of data lost can result in financial catastrophe for the customer and the bank.
- A publicly-traded company loses the work it has done to prepare its mandatory quarterly filings. Whenever a company misses a mandatory filing date, there are significant financial and procedural penalties, including de-listing the company and criminal charges against its officers.
- A research scientist stores unfiled patent documentation on his laptop computer. He also backs up the data to a secured server. If these documents are modified by anyone other than the user, or if the data is destroyed, this significant research investment may be lost.

We could go on for a long time with examples about the cost of data integrity or data loss. But this concept is relatively well understood to most BDMs. Most organizations that rely on data for their core business have implemented data categorization—identifying data that, if lost, has significant impact on the company. This is often called high-value data or high business impact data.

Special precautions are taken to ensure that high-value data is not lost or corrupted. These precautions usually start with regular, verified, and secured data backups. This helps ensure that the data can be recovered in case of loss. Because restoring data from backup can be costly and does not always restore the most current version, most organizations choose to take steps to help prevent the loss in the first place. Often this means that the organizations also employ a data defragmentation solution. This helps with both system stability (explored in the next section) and data integrity.

Stability

System stability directly results in decreased system cost. This concept is applicable to almost any capital asset. As an example, consider your car. It is a capital asset—you paid a significant price for it and expect it to work for many years. You may plan your career and personal life around the fact that you have a car. Of course, these plans most likely depend on the car working properly. Few people purchase a car and make plans based on it working 90% of the time. Even fewer people own a car and expect it to break down during a drive to the hospital or a job interview. In these cases, you might have to pay for a taxi or private car service to get to your destination, which would significantly increase your transportation costs.

The same paradigm applies to computer stability and cost. Unexpected downtime is incredibly expensive in a number of ways. The users' inability to use the system is the most obvious impact. But consider the expense of out-of-warranty repairs to a system. There is also the operations cost of identifying and replacing the failed system. In almost all cases, preventing system downtime is a better investment than repairing the system when it fails.

For example, the Plantronics Corporation conducted a small internal study. They compared the stability of desktop computers before and after running disk defragmentation software. Everything else remained the same—the system workload, the hardware and software configurations, and so on. When the system was defragmented, the users consistently reported that their systems were more reliable and performed better than before the defragmentation. Technical reasons aside (see Chapter 2 for the reasons), the system stability improved both the perception and reality of the system's reliability.

A similar case study comes from the Web hosting company CrystalTech, where disk fragmentation was decreasing system performance so severely that customers complained and some systems had to be taken offline for maintenance to defragment the disks. Similar to Plantronics, the implementation of a defragmentation solution improved both the real and perceived performance and uptime of the systems.

A more technical and less user-oriented case study was conducted by *Windows IT Pro* magazine in June 2007. When the researchers intentionally fragmented certain key components of the system, there was a significant decrease in system stability. However, as soon as they used a defragmentation solution to address the problem, the stability issues were resolved.

Cost-Benefit Analysis Summary

We've seen that disk fragmentation can be a significant problem. This isn't just a technical problem—it is also a financial problem as well as a business problem. Fragmentation can have a significant monetary impact on any organization. And its impact is not just a minor nuisance; it can significantly impact the entire business, even if the business is not technology-focused (as we saw with the Ruby Tuesday case). As we've explored, there is a wide span of problems that disk fragmentation causes and you should understand the necessity for a solution in your company. The next section describes how to choose the solution that's best for you and integrate it in your company.

How to Make Your Decision

You've looked at the available defragmentation solutions. You've decided on a default defragmentation method and potential exceptions. You have an idea of how many computers will receive the software and how it will be deployed. Now you need to make your purchase and use it.

The remaining phases of the purchasing process are fairly straightforward. These are common to any software evaluation decision:

- Preselection
- Test
- Purchase
- Deployment

Let's take a brief look at each of these phases from a BDM's perspective.



This section is similar to the identically-titled section in Chapter 3. Although much of the data is the same, it has been customized to be more useful from a business point of view instead of a technical implementation viewpoint.

Preselection

Now that you've identified the needs of your organization, take a look at the solutions available. There are a number of ways that you can find out information about the features of the software packages:


- Read marketing literature from the software developer
- Review industry-provided case studies
- Check software reviews from other corporate users, IT managers, and BDMs
- Visit the company's Web site
- Ask the manufacturer to have a sales representative contact you
- Network to find others who use the same software and ask them their opinions

The desired result of this work is that you'll have one or two solutions that you believe will work best for your needs. There may be a long list of potential solutions, but using these methods against the decision criteria we developed earlier should help bubble the best candidate to the top. Once that happens, we can examine the best candidate through testing.

Test

Testing any significant IT investment is a critical step in the selection process. No matter how much research you perform, you should see it work before you commit to the solution.

The BDM rarely performs any hands-on tests. But this person does need to ensure that the tests are being carried out by the IT team and that the data being gathered can be used to make a trustworthy business decision.

 The testing instructions in Chapter 3 are extensive and should provide a solid foundation for any testing process.

The results that you receive from the IT team doing the testing should include answers to the following questions:

- Does the software perform the functions that it advertises?
- Does it address the three categories of issues discussed in this guide?
- Is the software reasonably easy to deploy and manage?
- Did the software affect any other business software or systems? Were there any conflicts?
- Was the software tested in the manner it would be implemented on production systems (in real-time, on a schedule designed for production systems, etc.)?

These answers will not necessarily provide a complete picture of the solution or drive you towards a single product. Instead, the test results should be balanced with other decision-making criteria that apply to any IT purchase such as price, supportability, and long-term value.

Deployment Guide as a Result of Test

Most organizations overlook one key element of testing that often justifies the entire process. During testing, you have to deploy and redeploy a number of times. And you're documenting the process as you go. A natural result of this work should be a Deployment Guide for the software that you can use in production. This detailed guide will be fully tested and verified before the end of the test process. It is an invaluable document for your deployment staff because they can understand exactly what steps to perform, what results to expect, and how to handle any variances that may occur. And if you're doing a thorough job of testing, this document should require virtually no additional effort.

Once you've completed the testing and combined the results with the other information you have, you should have enough information to decide whether to proceed with the purchase and widespread deployment of the software. But do not be surprised at this point if the project takes a different direction. The results of applied testing sometimes help us draw different conclusions than we had previously thought. For example, you might find that your preselected and tested defragmentation solution conflicts with a disk management application that you use on 25% of your computers. In that case, you would be unable to proceed with the deployment, at least to those affected computers. You might decide not to deploy any defragmentation solution to those computers, to use two defragmentation solutions, or to test your second choice to see if it also has the same issue. But obviously it's better to find this out before you've purchased licenses and begun your widespread deployment.

Purchase

The purchase process will be different for every customer and every software vendor. Virtually every purchase is going to vary to some degree. So providing specific details here isn't really useful. Some software vendors will negotiate bulk pricing, while others will not. Some will accept purchase orders or incremental purchases at the same discount, others will not. You might receive your software funding over time or all at once. The possibilities are endless.

You should ensure that you have ready access to the software. Receiving some number of retail shrink-wrapped packages is one solution. These are effective as known good, clean copies for building system images and performing test installations from local media instead of the network. You should ensure that you have enough on hand in case of problems like the loss of a software deployment server or having to install the software to an isolated or offline system. Some software companies offer alternative methods for software acquisition and storage, such as online libraries or the option to burn their software to CD/DVD on demand. Use whatever method you're comfortable with, as long as you have access to a backup of the software in case of emergency.

Deployment

Great! You've analyzed the market, selected a software package, tested it thoroughly enough to know it works for you, and purchased enough licenses to begin your deployment. Now let's get going!

The section on deployment in Chapter 3 covered the majority of deployment considerations and decision criteria that you make. But by the time you get to this stage, you almost certainly have a very specific deployment strategy, plan, and documentation. Now it is time to execute on your well thought out and documented strategy.

In a perfect world, the deployment is the easiest part of the process. But in reality, issues will arise. Conflicts will come to the surface that weren't detected during testing. Your deployment software might hiccup and miss a hundred users. The new software might conflict with another application that's only deployed on a small number of computers, so you missed it during testing. Regardless of how well you planned, remain flexible and deal with the snags as they arise.

Consider Standardizing and Automating Your Deployments

If your organization does not currently have a standardized deployment strategy, you should consider investigating this option. The benefits to having one are almost too numerous to list but include ensuring IT consistency across the organization, more effectively managing software licenses, and reducing the total cost of ownership (TCO) of systems by reducing the deployment time of a computer from hours or days down to minutes with little or no administrator interaction. Consider reviewing Microsoft's [Business Desktop Deployment 2007](#), which includes both guidance and automated tools and is available for free download.

At the end of the deployment phase, you have your solution installed and running on all the intended computers with the software verified and reporting its status. But deployment isn't really ever complete. New computers come into the environment and require one-off deployments. Old computers require undeployment or reconfiguration. This is part of the ongoing software operation life cycle, but it is the same as any other piece of software.

Summary

Disk fragmentation is a serious problem that affects every business that relies on computer systems. Even companies that don't focus on technology can be severely impacted, but often in less obvious ways. System crashes and errors may be one very apparent symptom of disk fragmentation. Less obvious symptoms are loss of productivity, both employee-based and computer-based.

If it isn't already apparent, let's be very clear: you should evaluate and implement a disk defragmentation solution in your company. It doesn't matter if you have 50 computers or 50,000. Fragmentation is almost certainly causing some negative impact on your organization. You should use the techniques described in this guide to determine the proper solution for your company and then integrate it to all your computers.

Glossary

Bottleneck

A part of a process that causes the entire process to slow.

CD-ROM File System (CDFS)

An older file system used to store data on a compact disc.

Cluster

The basic disk allocation unit for a file system. It is made up of one or more physical disk sectors.

Compression

In NTFS, attempting to make files take less room on a disk by compressing data in the file. This is also called file compression.

Defragmentation

The act of reallocating data storage on a disk such that each file's data is located in as few contiguous data runs as possible.

ext3

A file system normally associated with the Linux operating system.

Extent

A physically contiguous group of disk clusters used by a file as a single storage unit. Also called a *run*.

File Allocation Table (FAT)

An older file system most closely associated with MS-DOS and early versions of Windows. FAT came in several varieties depending on the disk addressing scheme: FAT12, FAT16, and FAT32. FAT is still in use on some older systems or where MS-DOS compatibility must be maintained.

File system

A standardized scheme for storing and retrieving data from secondary storage. Common file systems include NTFS, CDFS, and FAT.

Fragmentation

The state in which data is no longer stored contiguously. This can be file fragmentation, where the file on the secondary storage device is stored in discontinuous extents, or data fragmentation, where the data structure within the file is discontinuous or suboptimal.

High Performance File System (HPFS)

An older file system primarily used on OS/2 and Windows NT systems where OS/2 compatibility is required.

Integrated Drive Electronics (IDE)

An interface specification for connecting storage devices (such as hard disks) to computers.

Metadata

Data structures that describe the data stored in an NTFS volume. Metadata in NTFS contains several different files and data structures, most notably the MFT.

Master File Table (MFT)

The special metadata file on NTFS volumes that describes where all the resources on that NTFS volume are located, such as which clusters contain which directories and files.

New Technology File System (NTFS)

A transactional file system that supports metadata storage. Most newer Windows systems use NTFS as their default file system.

Pagefile

A file on secondary storage dedicated to storing data temporarily transferred (or paged) from main memory. A pagefile acts as a sort of extended RAM.

Run

See *extent*.

Serial Advanced Technology Attachment (SATA)

A newer version of the IDE disk interface standard that offers faster throughput and simpler connection options than IDE.

Small Computer System Interface (SCSI)

A popular disk interface best recognized for its high throughput and flexible disk connection options.

Secondary storage

Long-term persistent data storage device, most commonly implemented as a hard disk. This differs from primary storage, which is usually a short-term fast but volatile storage method such as RAM.

Sector

The basic unit of hard disk storage. Normally sectors are not addressed individually but are grouped into clusters by the file system.

System files

A specific type of file used by the operating system to provide critical elements of system functionality. System files are often treated specially by the file system.

Temp files

Temporary files created by many applications during normal data processing applications. Temp files are usually created for a short period of time and then deleted.