Realtime
publishers
"Leading the Conversation"

The Shortcut Guide™ To

Managing Disk Fragmentation

sponsored by

Diskeeper
corporation

Mike Danseglio

Copyright Statement

[**Editor's Note:** This eBook was downloaded from Realtime Nexus—The Digital Library. All leading technology guides from Realtimepublishers can be found at http://nexus.realtimepublishers.com.]

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 build-up, 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.

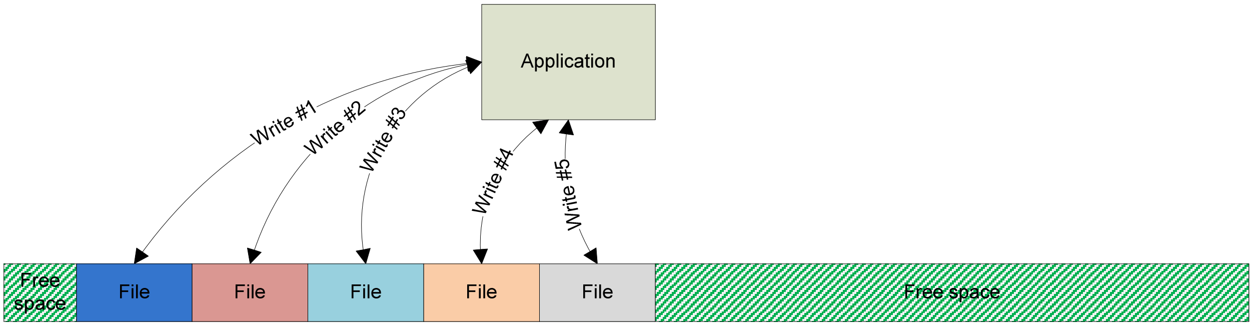

Figure 2.1: An application creating a number of small contiguous files on a hard disk.

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.

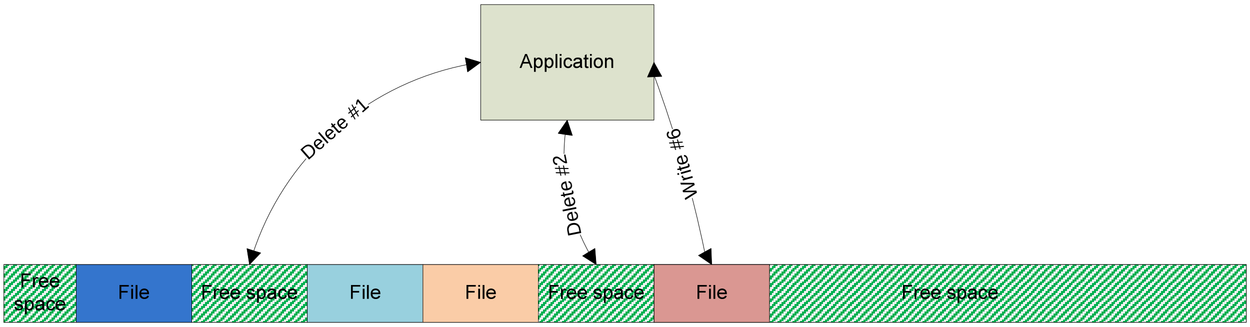

Figure 2.2: An application during normal operation, deleting some files, updating others, and writing new ones.

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.

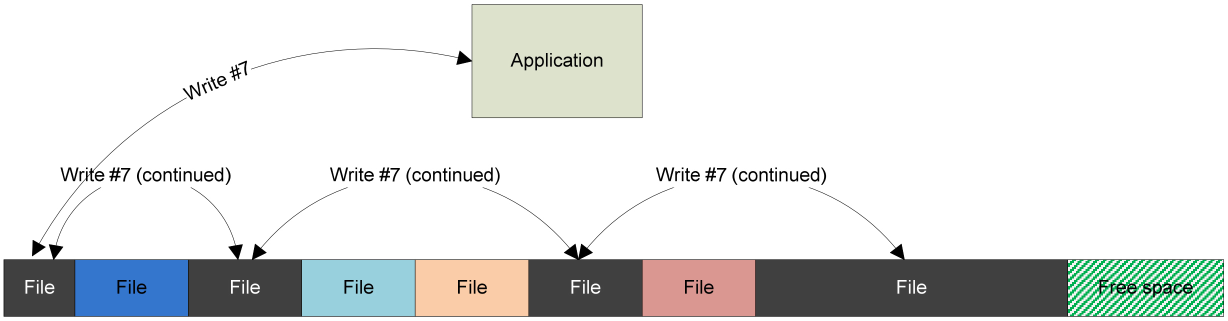

Figure 2.3: Writing a big file to the hard disk while the free space is fragmented.

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.

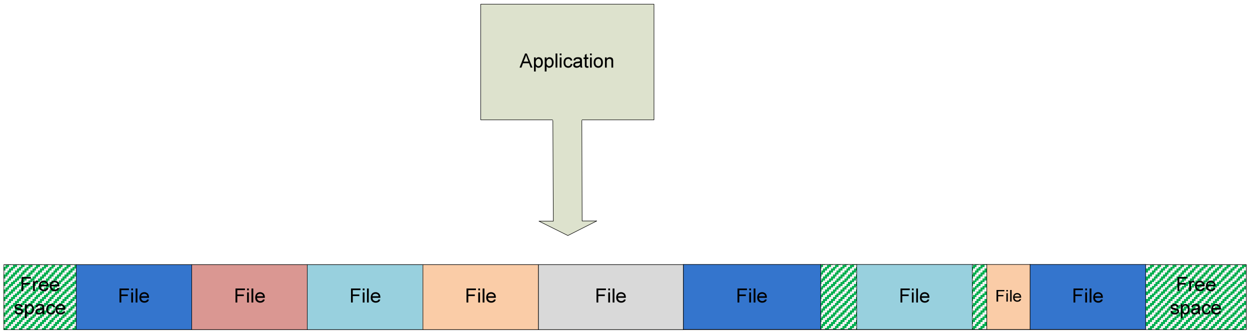

Figure 2.4: With such sparse free space, the application will almost certainly create a fragmented file.

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.

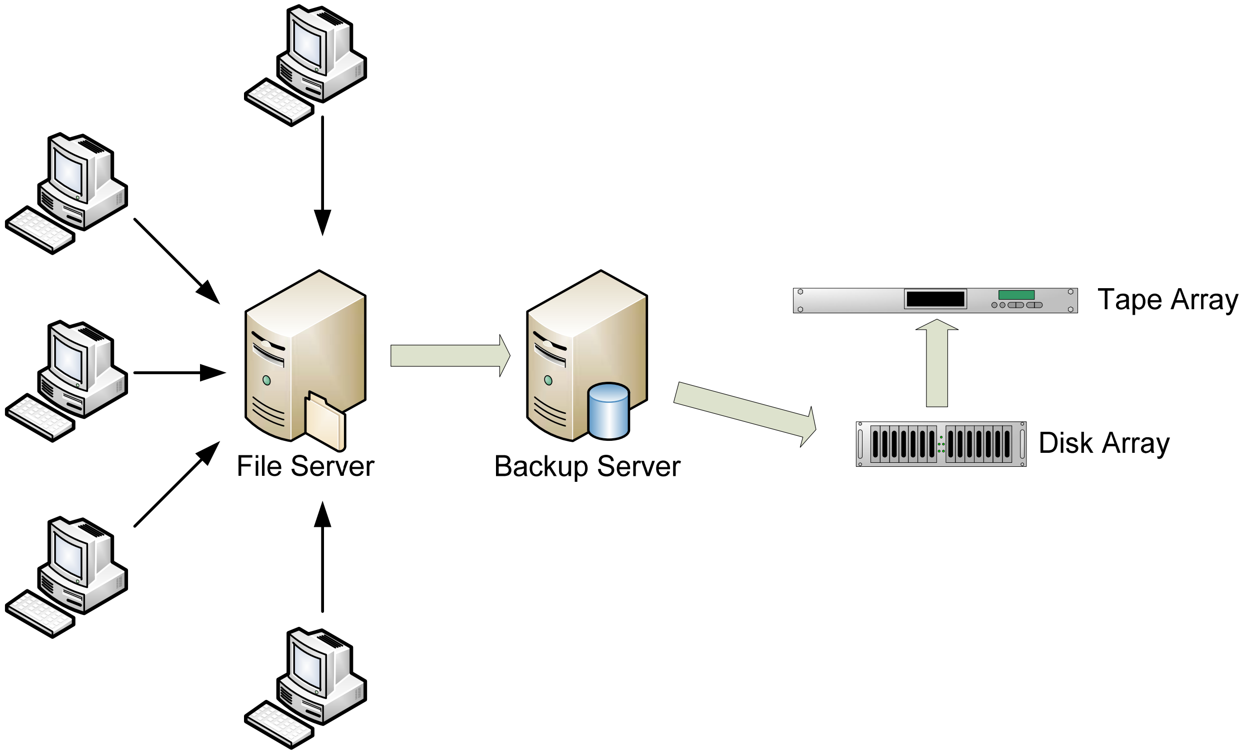

Figure 2.5: The data flow for a D2D2T backup.

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.

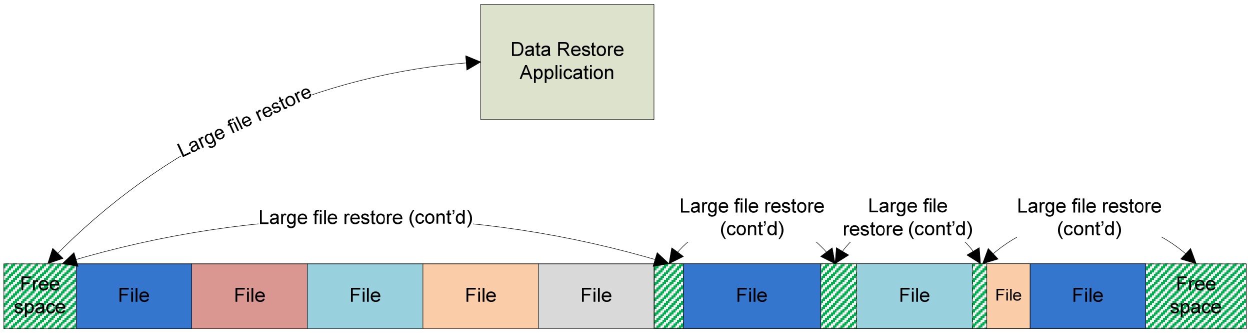
Data Restore Application
Large file restore
Large file restore (cont'd)
Large file restore (cont'd)
Large file restore (cont'd)
Large file restore (cont'd)
Free space
File
File
File
File
File
File
File
File
File
Free space

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.

Download Additional eBooks from Realtime Nexus!