Windows IT Pro

The Impact of Disk Fragmentation

by Joe Kinsella

While there is little dispute among IT professionals regarding the impact of disk fragmentation on system performance, no independent guidelines exist to recommend the frequency of defragmentation across an infrastructure. Some IT professionals use defragmentation as a measure of last resort, defragmenting only after system performance has sufficiently degraded to make its impact directly noticeable to users. Others proactively schedule disk defragmentation regularly, with the intent of eliminating the gradual accumulation of fragmented files.

While just about every IT professional has his or her fragmentation horror story – about a system on which fragmentation had so severely degraded performance that it was unusable – very few can offer more than anecdotal evidence regarding the use and importance of defragmentation software.

I recently decided to put fragmentation to the test, with the intent of answering two basic questions:

1. What impact does fragmentation have on user and system activities?
2. How quickly does fragmentation accumulate as a result of these activities?

This white paper will outline the results of the testing, draw conclusions, and make recommendations regarding managing fragmentation across your infrastructure.

Contents

Diskeeper is now

Condusiv™ Technologies

The Impact of Disk Fragmentation

The Mechanics of Disks

The basic components of hard disks (see Figure 1) have not changed significantly since their invention in the 1950s. Hard disks have one or more polished platters made of aluminum or glass that hold a magnetic medium used for storing information. The platters are stacked onto a spindle and rotated by a spindle motor at very high speeds, often in excess of 160 miles per hour. A platter has concentric circles called tracks, and each track is divided into small sections called sectors, each capable of holding a fixed amount of information.

Small devices called heads are responsible for the actual reading and writing of data on the platter. Each platter has two heads (for the top and bottom), and the heads are mounted on sliders positioned over the surface of the disks, which in turn are mounted on arms. The entire assembly is connected to and controlled by an actuator, which in turn is connected to a logic board that allows for the communication between a computer and the hard disk.

To read or write information to the disk, an application makes a request of an operating system to create, modify or delete a file. The operating system then translates the logical request into a physical request containing the actual locations to be read or written on the hard disk. The logic board then instructs the actuator to move the heads to the appropriate track, and to read or write the appropriate sectors from the rotating platter below.

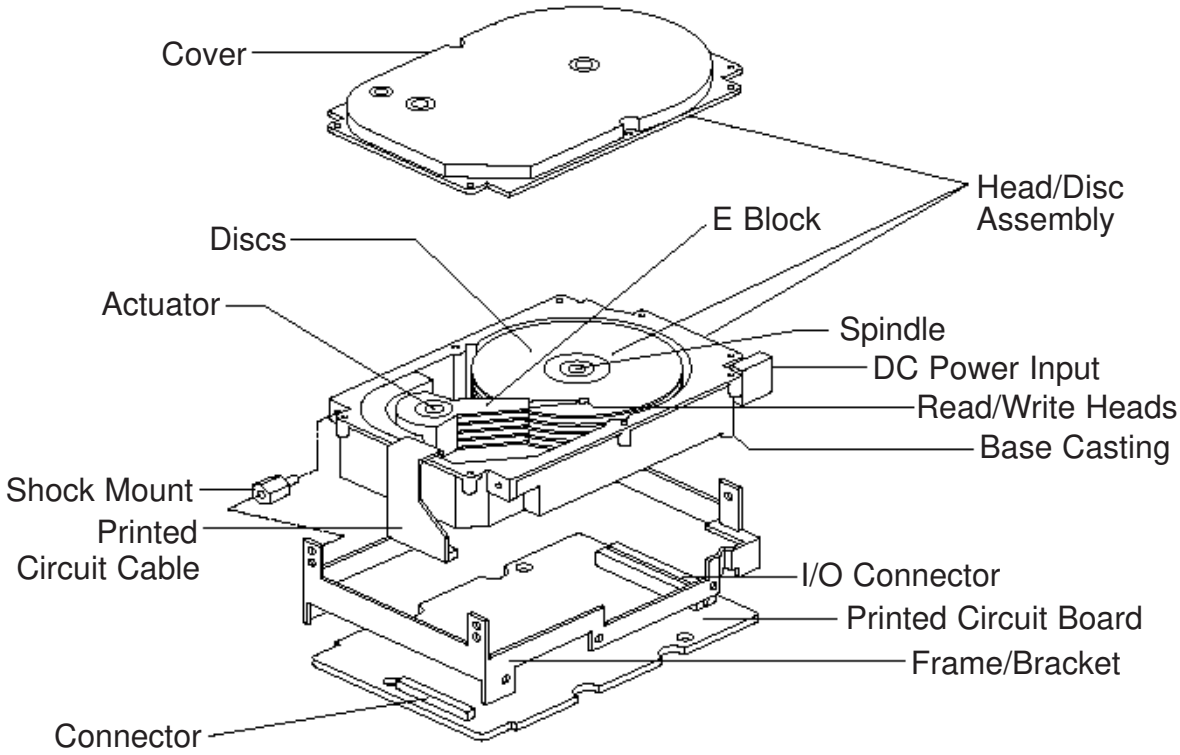

Figure 1: Overview of Hard Disk

The mechanical movement of the head across a platter is typically one of the most expensive operations of a hard disk. As a result, most operating systems seek to minimize this head movement through caching, optimizing I/O requests, and streamlining the storage of data on a disk. Streamlining the storage of data typically involves writing the data for individual files in a file system contiguously on a platter, allowing the head to read or write data without needing to be repositioned.

Due to their mechanical nature, hard disks represent one of the poorest-performing components in a system. Electronic components, such as the CPU, motherboard, and memory, are improving performance at a much faster pace than hard disks, whose performance is limited by the mechanics of spinning a platter and moving a head. As a result, since an integrated system is often as fast as its slowest component, it is essential to ensure hard disks are performing at their optimum level.

NTFS

Before discussing fragmentation, let's discuss briefly the New Technology File System (NTFS), the file system used with all modern versions of Windows and the focus of the testing for this white paper. While understanding a specific file system is not a pre-requisite to understanding fragmentation, it will help clarify both the terminology used as well as the test results.

NTFS was created by Microsoft in the 1990s as part of its strategy to deliver a high-quality, high-performance operating system capable of competing with UNIX in a corporate environment. NTFS divides a hard disk into a series of logical clusters whose size is determined at the time the disk is formatted with the file system. A newly formatted hard disk will by default be formatted with 4 KB clusters.

The cluster size is important because it determines the smallest unit of storage used by the file system. This means that a 1-byte file on a hard disk formatted with NTFS with a 4K cluster size will actually physically take 4K of space on the disk (which is why Windows reports both the *Size* and *Size on disk* for all files).

The file system is divided into two parts: the Master File Table (MFT) and a general storage area. You can think of the MFT as the table of contents for a hard disk. The MFT contains a series of fixed-sized records that correspond to a file or directory stored in the general storage area. The information captured in MFT records is called attributes, and includes such information as the name of the file, its security descriptors, and its data. Two types of attributes

are in an MFT record: resident and non-resident. Resident attributes reside within the MFT. Non-resident attributes reside in the general storage area. If the amount of space required for all the attributes of a file, including its data, is smaller than the size of the MFT record, the data attribute will be stored resident. Because a record size is typically the same as the cluster size, only very small files will be entirely resident within the MFT. Most files contain non-resident attributes in the general storage area.

Upon formatting a disk, the first 12% of space is assigned to the MFT, and the remaining 88% is allocated to the general storage area. As more files and directories are added to the file system, NTFS may need to add more records to the MFT. In doing so, NTFS will allocate space from the general storage area to the MFT.

The Cause of Fragmentation

When a file is stored in clusters that are not physically located next to each other on the platter, it is fragmented. Fragmentation can occur for various reasons, but the most common cause is the modification or deletion of files. For example, if you deleted a non-fragmented 40K file that occupied 10 contiguous clusters on an area of the disk surrounded by other used clusters, the disk will now have 10 free clusters available for use. If you then saved an 80K file, which requires 20 clusters, the operating system may choose to use the 10 recently free clusters and then find an additional 10 clusters from somewhere else on the disk. This means our 80K file is now fragmented, residing in two different locations on the disk.

Over time, files in NTFS tend to be broken into more and more non-contiguous clusters on a disk. This fragmentation of files accumulates over time, causing a gradual increase in the movement of a head across the platters of a hard disk, and thereby resulting in a gradual increase in the time for each I/O operation.

The impact of fragmentation on system performance differs based on the usage of the fragmented files. For example, a single infrequently used Microsoft Office document is unlikely to have an impact on overall system performance. However, fragmentation of a paging file, which provides virtual memory to all applications on a system, will likely have a more noticeable impact.

Fragmentation can affect all files, including system files. Fragmentation also can occur both in the MFT and in the general storage area. As the MFT expands to meet the growing number or files or directories, it can take over non-contiguous clusters, and thereby become fragmented. In addition, even the metafiles within the MFT can be allocated non-contiguous clusters and therefore be fragmented.

A generally repeated belief is that NTFS is resistant to fragmentation. Unfortunately, this is a myth. The underlying algorithm for identifying free space appears to readily re-use smaller non-contiguous space when in fact contiguous space does exist elsewhere on the disk. As a result, fragmentation will impact all Windows systems.

The Approach to Testing

To quantify the impact of fragmentation, I ran tests using typical user and system activities on a computer running Windows XP Professional. I specifically focused on word processing, email, Web browsing, anti-virus and anti-spyware applications

The first challenge I needed to solve to ensure accuracy of my testing was to be able to simulate the natural fragmentation that would occur on users' hard drives. I could not rely on actually fragmented hard disks for two reasons. First, since no two systems are fragmented in exactly the same way, it would not be possible to test different levels of fragmentation with naturally fragmented systems. Second, since my tests focused on specific applications, I need to isolate the fragmentation to the application under test, and not have fragmentation in other areas of the disk (e.g., the MFT or page file) affect my test results.

My solution to this challenge was Simfrag.exe, a utility that can fill free disk space with files equal to the cluster size, and then remove files at specified intervals. The usage and then removal of files produces pre-determined patterns of used and unused clusters that allowed me to achieve greater consistency in my tests. It also allowed me to control the location of the fragmentation, ensuring that the use of any free space would equally impact newly created files.

To limit the impact of Simfrag.exe to the free space on a computer, the hard disk was fully defragmented before running it. It is important to note that a system on which Simfrag.exe was run is not actually fragmented, but instead has used clusters spread in patterns across the hard disk, rendering any newly created files on the disk with a higher likelihood of fragmentation.

To test the applications at different levels of fragmentation, I ran my tests on the same system but with different images. These included baseline, low, medium and high fragmentation images. The primary difference between each image was the ratio of the used to unused clusters in the free space produced by SimFrag.exe. For example, the low fragmentation image had a 1:10 ratio of used to unused

	Baseline Image	Low Fragmentation Image	Medium Fragmentation Image	High Fragmentation Image
Ratio of used to unused clusters in free space	0:0	1:10	1:5	1:1
% Free space	61	43	35	23
Total files	10522	60351	135365	260357
Average file size (KB)	155	27	14	10
Total directories	676	1655	1664	1657

Figure 2: Overview of Images

clusters, meaning that for every used 4K cluster in its free space, there was a contiguous gap of 9 unused 4K clusters. For additional detail about the images, see Figure 2.

The actual fragmentation in my testing results from the setup for each test. Each test begins with an action that results in the creation of a number of new files on the disk (e.g., copying files, retrieving Web pages). As an example, the test setup for Microsoft Word requires the copying of 100 MB of Word documents to the disk. The purpose of the test setup is to cause fragmentation in the newly created files that allow me to assess the impact of fragmentation on a specific application.

All testing was performed on a 3.0 GHz P4 with 256 MB RAM, and a 20 GB hard disk (7200 RPM, 8 ms average seek time) formatted using a 4K cluster size. The operating system for the tests was Windows XP Professional. Each test was performed with multiple iterations based on a predefined test plan, and the results published here represent the average of these runs.

The testing focused on the impact of fragmentation on software applications and data, and not on the overall system. The tests selected are intended to reflect the types of user and system activities on a typical Windows desktop in a corporate environment.

The Impact of Fragmentation

To assess the impact of fragmentation, I ran a series of tests using Microsoft Word, Microsoft Outlook, Microsoft Internet Explorer, Microsoft Anti-Spyware, and Grisift AVG Anti-Virus Scanner. Each test started with a fresh image restored to the disk using imaging software. Files required for the test were then copied to the disk, which resulted in fragmentation corresponding to the different types of images. For a summary of the test results, see Figure 3.

Microsoft Word

I ran three tests on Microsoft Word: testing the load of a large document, save as of a large document, and document searching. These tests were run against the baseline, low, medium and high fragmentation images. Before starting the test, I copied 100 MB of Microsoft Word documents (220 files) to the My Documents folder.

I found that the performance impact of fragmentation on Microsoft Word ranged from 5% to 1489%. Loading a large Word document took 17.7 seconds with the baseline image, but took over 50.9 seconds on the high fragmentation image (+187%). Saving a large Word document took 2.9 seconds on the baseline image, but over 46.1 seconds on the high fragmentation image (+1489%). And searching for documents matching specified text within the My Documents folder took 16.9 seconds on the baseline image, but over 70.4 seconds on the high fragmentation image (+68%).

Conclusion: Disk fragmentation can have a very severe impact (+1489%) on the performance of Microsoft Word.

Microsoft Outlook

I ran three tests on Microsoft Outlook: searching for emails in a folder, archiving messages in a folder, and restoring messages from an archive. These tests were run against the baseline, low, medium and high fragmentation images. Before starting the test, I copied a 500 MB Outlook database to the system.

I found that the performance impact of fragmentation on Microsoft Outlook ranged from 18% to 288%. Searching for emails with matching text in a folder containing 10,000 messages took 46.3 seconds for the baseline image, but 77.8 seconds for the high fragmentation image (+68%). Archiving 10,000 messages in a single folder to disk took 241.5 seconds in a baseline image, but 378.4 seconds in a

Application	Test	Baseline Image	Low Fragmentation Image	Medium Fragmentation Image	High Fragmentation Image
Microsoft Word	Load 30 MB document (seconds)	17.7	18.4	28.3	50.9
Microsoft Word	Save 30 MB (seconds)	2.9	5.4	23.5	46.1
Microsoft Word	Document text search across 100 MB of Word documents (seconds)	16.9	29.4	53.0	70.4
Microsoft Outlook	Search for matching emails in folder (seconds)	46.3	95.1	78.5	77.8
Microsoft Outlook	Archive of messages in a folder (seconds)	241.5	288.5	285.4	378.4
Microsoft Outlook	Restore of messages from an archive (seconds)	52.3	116.0	106.1	140.0
Microsoft Internet Explorer	Retrieve non-cached Web pages (seconds)	10.4	27.2	19.1	37.0
Microsoft Internet Explorer	Retrieve cached Web pages (seconds)	9.8	15.9	24.2	26.5
Grisoft AVG	Anti-virus scan of My Documents (seconds)	48.9	175.5	112.3	215.5
Microsoft Anti-Spyware	Anti-spyware scan of My Documents (seconds)	64.5	144.3	112.3	87.3

Figure 3: Impact of Fragmentation

high fragmentation image (+56%). And restoring an archive containing 10,000 messages took 52.3 seconds in the baseline image, but 140.0 seconds in the high fragmentation image (+167%).

Conclusion: Disk fragmentation can have a severe impact (+288%) on the performance of Microsoft Outlook.

Microsoft Internet Explorer

I ran two tests on Internet Explorer: surfing to 10 cached Web pages, and surfing to 10 non-cached Web pages. These tests were run against the baseline, low, medium and high fragmentation images. To minimize the impact of network latency on the test results, all Web pages were retrieved from a local Web server, and were driven by an automated program to eliminate human error. Before starting the test, I filled the Internet Explorer cache with 90 MB (118,000) of temporary files. Each Web page retrieved in the test was identical and was comprised of 170K of both HTML and assorted images.

I found the performance impact of fragmentation on Internet Explorer ranged from 14% to 198%. Retrieving 10 Web pages that were in the Internet Explorer cache took 9.8 seconds for the baseline image, but 26.5 seconds in the high fragmentation image (+90%). Retrieving 10 Web pages that were not in the Internet Explorer cache took 10.4 seconds in the baseline image, but 37.0 seconds in the high fragmentation image (+198%).

Conclusion: Disk fragmentation can have a severe impact (+198%) on the performance of Microsoft Internet Explorer.

Microsoft Anti-Spyware

I ran a single test using Microsoft Anti-Spyware, the soon to be released free anti-spyware software. I ran a full anti-spyware scan of the system, but limited its file scanning to the My Documents folder. Before starting the test I copied 500 MB of assorted documents to the My Documents folder.

Application	Test	Baseline Image (seconds)	Change From Baseline After Defragmentation (%)	High Fragmentation Image Before Defragmentation (seconds)	High Fragmentation Image After Defragfrag-mentation (seconds)	Change After Defrag (%)
Microsoft Word	Load 30 MB document	17.7	9.0	50.9	19.3	-62.1
Microsoft Word	Save 30 MB	2.9	58.6	46.1	4.6	-90.0
Microsoft Word	Document text search across 100 MB of Word documents	16.9	17.8	70.4	19.9	-71.7
Microsoft Outlook	Search for matching emails in folder	46.3	16.6	77.8	54.0	-30.6
Microsoft Outlook	Archive of messages in a folder	241.5	18.0	378.4	285.0	-24.7
Microsoft Outlook	Restore of messages from an archive	52.3	43.2	140.0	74.9	-46.5
Microsoft Internet Explorer	Retrieve non -cached Web pages	12.4	2.4	37.0	12.7	-65.7
Microsoft Internet Explorer	Retrieve cached Web pages	13.9	-12.2	26.5	12.2	-54.0
Grisoft AVG	Anti-virus scan of My Documents	48.9	19.2	215.5	58.3	-72.9
Microsoft Anti-Spyware	Anti-spyware scan of My Documents	64.5	-5.9	87.3	60.7	-30.5

Figure 4: Performance Change After Defragmentation For High Fragmentation Image

I found the performance impact of fragmentation on Microsoft Anti-Spyware ranged from 35% to 123%. A spyware scan took 64.5 seconds in the baseline image, but 87.3 seconds in the high fragmentation image (+35%). A spyware scan took 144.3 seconds in the medium fragmentation image (+123%).

Conclusion: Disk fragmentation can have a significant impact (+123%) on the performance of Microsoft Anti-Spyware.

Grisoft AVG Anti-Virus Scanner

I ran a single test using AVG Anti-Virus Scanner. I ran an anti-virus scan of the My Documents folder. Before starting the test I copied 500 MB of assorted documents to the My Documents folder.

I found the performance impact of fragmentation on anti-virus scanning ranged from 112% to 340%. An anti-virus scan took 48.9 seconds in the baseline image, but 215.5 seconds in the high fragmentation image (+340%).

Conclusion: Disk fragmentation can have a severe impact (+340%) on the performance of AVG Anti-Virus scanner.

The Impact of Defragmentation

To quantify the impact of defragmentation on a system, I defragmented each hard drive using Diskeeper, a leading

Test	Number of Fragmented Files	Number of Excessive File Fragments	Average Fragments Per File
Install Windows XP Professional	439	1926	1.20
Install Microsoft Office	34	382	1.03
Install Windows XP Service Pack 2	197	1656	1.09
Installing 16 critical Windows updates	157	621	1.05
Retrieving 1500 Web pages	2396	6178	1.27
Copying 500 MB of files	24	143	1.01

Figure 5: Accumulation of Fragmentation

commercial defragmentation application, before re-running each of the previous tests. The performance improvements varied from test to test and image to image, but in general showed moderate to significant performance gains.

Figure 4 shows a summary of the impact of defragmentation on the high fragmentation image. The results in this table were typical of the results from the other images. The *Change After Defrag* column provides the percentage performance improvement gained by simply defragmenting the existing image (note: negative numbers represent performance gains, positive numbers performance loss). The change in performance from defragmentation ranged from -30% to -90%. The *Change From Baseline After Defrag* provides the percentage change in performance between the defragmented high image and the baseline image. The change in performance ranged from +58% to –12%. On average there was a slight drop in performance between the defragmented high image and the baseline, but in general the performance between the two images is roughly comparable.

The Accumulation of Fragmentation

After collecting evidence of the negative impact of disk fragmentation on the performance of a system, and validating the performance gains from defragmenting a disk, I ran six tests to better understand the speed with which fragmentation accumulates on a system. The six tests included installing an operating system, installing application software, installing service packs, installing critical hotfixes, surfing the Web, and copying files.

Each test started with a fresh, fully defragmented baseline image to ensure any resulting fragmentation was directly the result of the test performed, and not already resident on the disk (note: Simfrag.exe was not used for these tests). Figure 5 shows the results of these tests.

The results of the tests demonstrate the speed with which fragmentation can occur on a hard drive. Installing Windows XP Professional resulted in the fragmentation of 439 files, including important files from the system directory and the DLL cache. The most fragmented file was a single 22K file that was broken into 360 fragments across the disk.

Installing Microsoft Office resulted in the fragmentation of 34 files, across both application and supporting files. The most fragmented file was the primary Microsoft Excel executable (EXCEL.EXE), which was a 7 MB file divided into 67 fragments across the disk.

Installing Windows XP Service Pack 2 resulted in 197 fragmented files, mostly across log and temporary files. The most fragmented file was WINDOWS\system32\config \system.log, which was a 1K file broken into 110 separate fragments.

Installing 16 critical Windows updates resulted in the fragmentation of 157 files across a variety of both log and system files. The most fragmented file was WINDOWS \system32\config\system.log, which was a 1K file broken into 376 separate fragments.

Running an automated program to drive Internet Explorer to visit 1500 Web pages on the Internet resulted in 2396 fragmented files, primarily among Temporary Internet and System Volume Information files. The most fragmented file was a 14 MB System Volume Information file that contained 357 separate fragments.

Copying 500 MB of files (1800 files) on to the local hard disk using Windows Explorer resulted in the fragmentation of 24 files within the files that were being copied. The most fragmented file was a 90K file that resulted in 70 separate fragments. This test produced a particularly unusual result, because there was more than sufficient contiguous space on the disk for the 500 MB of files, but yet some files were still fragmented in the copy.

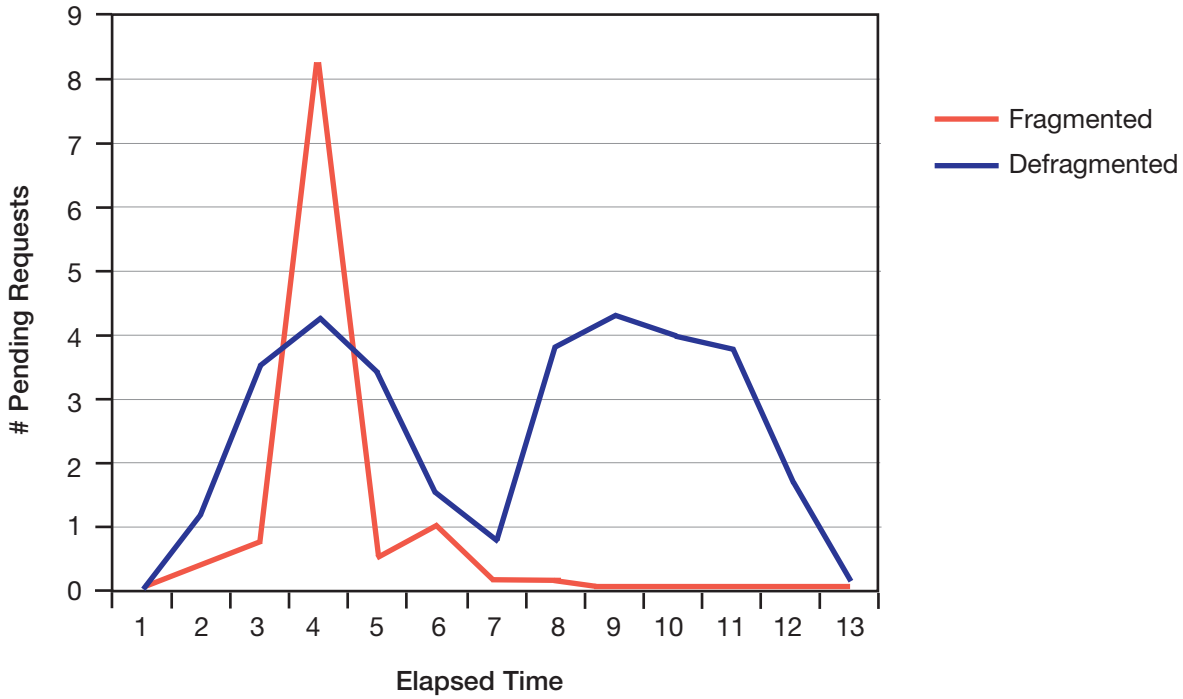

Figure 6: Disk Queue Length

Underneath the Hood

A properly defragmented disk drive will have lower and more predictable disk activity than a fragmented drive. Figure 6 compares the disk queue activity resulting from the copy of 100 MB of files to fragmented and unfragmented disk drives. The disk queue represents the number of pending requests to be sent by the operating system to the disk controller. While a high disk queue number is acceptable for bursts of disk activity, an average of less than two pending requests is generally accepted as indicative of well-performing systems. In my test, the disk queue for a defragmented disk quickly backs up eight and then immediately drops back down to less than one for the duration of the operation. However, the disk queue for a fragmented disk rises to four, drops back below one, rises again to four and then drops back down to less than one.

An explanation for this is the increased number of disk requests required to write data to a fragmented drive. The creation of a file on a defragmented drive should result in writing to a series of contiguous clusters on the disk. The creation of a file on a fragmented drive, however, requires writing to multiple non-continuous clusters on the disks that are likely to be spread across different platters and tracks. As a result, the operating system is more likely to break an I/O request to the disk into multiple requests as a result of fragmentation.

Figure 7 shows the breaking of input/output (I/O) requests into multiple requests for the same copy of 100 MB of files to both fragmented and defragmented disks. Split I/O can occur naturally as a result of operations on large files, but in general is indicative of fragmentation. There is a small amount of split I/O at the beginning of the copy (likely the result of one 30 MB file), but this quickly drops off for the duration of the operation. However, on a fragmented drive, the split I/O rises to over 250 requests per second, drops back to zero, rises over 250 again, and then drops back off. The split I/O graph has a very similar cycle to the graph of disk queue length.

Recommended Policy

In general, two types of defragmenters are available today: automatic and manual. Manual defragmentation software, such as the Disk Defragmenter built into Windows, needs to be run during periods of time in which the target system can be made unavailable. Automatic defragmentation software is designed to throttle its impact on a computer based on general system activity, and can be run without requiring that a system be made unavailable. It minimizes

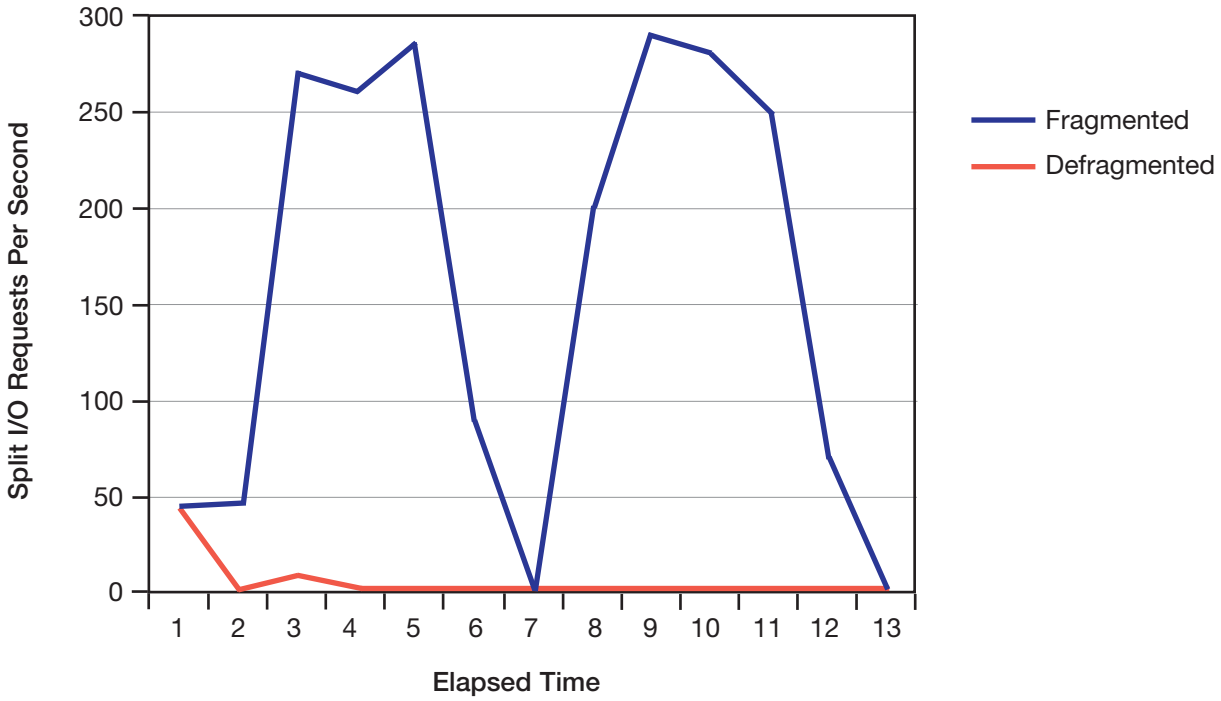

Figure 7: Split I/O

the administrative cost of managing defragmentation across your infrastructure, and can be considered a "set it and forget it" application. Automatic defragmentation software is therefore frequently used for systems requiring high availability and performance, or to minimize the administrative overhead of managing disk fragmentation across an infrastructure.

Defragmentation software also can be classified as either designed for the enterprise or designed for stand-alone usage. Defragmenters designed for the enterprise generally include features for centralized management, such as scheduling, reporting and alerting. Defragmenters designed for standalone usage do not have centralized features, and require that the software be run and the results reviewed on each system.

The demands on hard disks in a corporate environment have steadily increased over the last several years, due in a large part to increased Web surfing, increasing usage of host-based security applications (e.g. anti-virus, anti-spyware), and the increased application of hotfixes and service packs. As a result, defragmentation is increasing as a problem affecting the performance and availability of corporate computers, making it increasingly important to have a policy to manage it.

Based on my experience in the lab, I strongly recommend a proactive approach that includes daily defragmentation for critical systems (or systems requiring peak performance), and weekly defragmentation for all non-critical systems. In addition, I strongly suggest the use of automatic defragmentation software in order to minimize the administrative cost of managing and the downtime required to implement your defragmentation policy. For small environments, you will likely find the use of stand-alone defragmentation software to be sufficient. Medium to large environments are best managed through enterprise defragmentation software.

As a general rule, more frequent defragmentation will reduce both the time and system resources required to maintain performance.

Conclusions

Hard drives will fragment over time as a result of normal user and system activities. Some activities, such as the installation of applications or service packs, can result in the rapid fragmentation of a disk. Other activities, such as surfing the Web or using office applications, can result in the gradual accumulation of fragmentation over longer periods of time.

As fragmentation accumulates, the performance of a system degrades due to increased disk activity. Fragmentation in system files, page files and the MFT can result in a general degradation across all user and system

activities. Fragmentation in applications and data files can result in the degradation of specific software applications.

In my lab, I was able to demonstrate that fragmentation has a moderate to very severe impact on common desktop activities such as word processing, email, Web browsing, anti-virus scans and anti-spyware scans. In some cases, the performance impact resulted in activities taking almost 15 times longer than the same activities on non-fragmented drives.

As a result, fragmentation should be managed proactively, using automatic defragmentation wherever possible, in order to maximize the performance and health of your desktops and servers. A well-defined policy toward fragmentation can minimize down time, maximize performance, reduce the total cost of supporting systems, and in some cases prolong the need for hardware upgrades.

Joe Kinsella (jkinsella@silverbacktech.com) is the director of development for SilverBack Technologies, an integrated IT and security monitoring software provider. He is a CISSP with expertise in application and systems development.