Realtime publishers

The Essentials Series: Why You Need to Defragment

You Need to Defragment!

sponsored by

Diskeeper corporation

by Greg Shields

Copyright Statement

You Need to Defragment!

Can we all agree that there's nothing more frustrating than a slow computer? You've probably experienced the following situations, because they can all be common to your daily interaction with the Windows operating system (OS):

- You need to finish that spreadsheet before heading home to dinner and family, but instead you're watching the hourglass tick by.
- Maybe it's a quick email check before boarding that flight, but you forego the opportunity because your laptop takes 4 excruciating minutes to boot.
- Or, you're stacked with meetings and PowerPoint charts but find yourself in a waiting pattern as you reboot that un-cooperating conference room PC.

In these and dozens of other situations, you're at the mercy of your computer's processing. When it doesn't perform to the needs of your daily workflow, it can feel like you're working for it instead of it working for you. In virtually all these scenarios, that computer's lack of performance can be directly impacted by its level of fragmentation.

Fragment-Less Is the Goal

Article one in this series outlined the problem of fragmentation. It explained how fragmentation is a naturally-occurring side effect of a computer's disk activity. As something that cannot be naturally stopped, disk fragmentation must instead be managed to keep its spread from slowing your processing.

To that end, there are a number of established best practices associated with managing defragmentation as well as tools that enable proactive defragmentation. Although not all solutions are created equal, smart organizations select those with the right set of capabilities which ensure fragment-less systems both in the desktop and the data center. One long-held mechanism to accomplish this relates to the window in which defragmentation can occur.

Continuous > Scheduled

Traditional defragmentation solutions offer options for scheduling the defragmentation "pass" on your systems. This pass needs to be scheduled to occur at off-hour intervals, as its impact on system resources can be dramatic. The reassembly of file and folder fragments tends to be of great impact to the file system as well as the disk subsystem as a whole. Its processing can require a substantial amount of processor and memory resources as the defragmentation pass completes. These resources are necessary due to the multi-step process associated with analyzing a disk drive, looking for files, and correctly assembling them into a logical order. Should these activities occur without proper resource throttling and poor scheduling, the defragmentation process itself can be a major impact on your users' experience.

Yet although this concept of scheduled defragmentation has been a de facto norm for many years, many defragmentation solutions today leverage an alternative approach to optimizing file structures. Eliminating the schedule altogether, these solutions instead opt for a continuous approach to finding and fixing fragments.

Consider how this alternative approach improves the entire process. Article one discussed how the sheer number of fragments grows dramatically as a computer system is used. Computers with larger numbers of writes and a greater count of files tend to have a larger quantity of fragments. Thus, once the time interval goes by between fragmentation passes, the defragmenter starts at a disadvantage: To return a volume to a defragmented state, it must "catch up to" and eventually "get ahead of" the data processing of the system.

This problem tends not to be as challenging with desktop systems. Users of these systems often don't use them 24 hours a day. Thus, a natural period exists when processing is low and defragmentation can catch up. However, scheduled jobs on desktops can be problematic when users don't leave those systems powered on during the scheduled interval. Depending on the solution available, a powered off system can either miss the defragmentation schedule or be forced to run it shortly after the system is powered back on—and the user is ready to make use of it again.

With servers, the problems surrounding this approach grow even more insidious. Imagine the typical file server or database server, which tends to process its workload during the business day. High resource use actions for servers—such as patching, backups, and defragmentation—tend to collect during the evening hours. The co-processing of these intensive tasks over the short period away from business hours can aggregate to dramatically increase the overall time to accomplish each.

Contrast this situation with the continuous approach. Here, a computer's file system is always being monitored by the defragmentation solution. When fragments appear, those fragments are handled almost immediately. Today's enterprise defragmentation solutions leverage the interstices between user requests to accomplish the defragmentation process. As a low-level service that occurs in combination with the file system's writes, this incremental approach ensures that your disks remain defragmented and highly optimized at all times.

Proactive > Continuous

Yet even this continuous approach remains a reactive band-aid to a never-ending problem. Defragmentation products that rely exclusively on even a continuous approach find themselves working to resolve a problem that could be best solved by ensuring it never happens in the first place. This modern "proactive" approach to defragmentation dramatically changes the ways in which fragments are managed by a computer system.

For example, consider the situation where a new file is added to a perfectly defragmented disk. Even though this disk is completely free of fragments, "holes" of free space tend to lie across multiple areas on the disk. When this new file is added, the computer's file system attempts to locate a hole of free space within which to store the file. Using native tools alone, that file system is likely to store the file into a hole that isn't quite large enough to store the entire contents of the file. Immediately, a fragment is created as the file's contents are spread across multiple holes.

Using the continuous approach, once the file system completes its write, it is the job of the defragmentation engine to locate and reposition that file (as well as others that surround it when necessary) into a location where it is no longer fragmented. Using this process, the defragmentation engine is constantly forced to react to poor decisions that are made by the file system. When that file later expands, this doubling of effort repeats itself all over again.

Contrast this situation with one where the defragmentation engine and the file system work together instead of at odds with each other. Using this approach, any new file can be automatically written to the system's disk in such a manner so that little or no fragmentation occurs. File writes and expansions are compensated for by the defragmentation engine with the support of the file system itself. In essence, when using the proactive approach, a computer's disk largely prevents file fragmentation at any point. Solutions that leverage the proactive approach accomplish the same goal of a fragment-less system but with far less effort, impact on system resources, and the assurance that most file writes can be done without fragmentation ever occurring.

Fragmentation Impacts Everything

Ultimately, the sole purpose of defragmenting a computer is to increase performance. That point has been repeated thoroughly in this series already. But what kinds of processes are impacted by fragmentation? What types of user activities can be improved by the implementation of effective enterprise defragmentation? The first set of areas worth reviewing relates to the individual desktops and laptops of your users themselves. Consider the following user activities that are improved through the assurance of an always-optimized file system:

- *Slow application and OS response time.* Testing using the PCMark performance benchmarking tool has shown that a fragmented file system can have a dramatic impact on desktop performance (Source: http://downloads.diskeeper.com/pdf/NSTL_20Tests_20Diskeeper_20vs_20Built_20In.pdf). The running of this tool generates a metric that aggregates overall system performance and is intended to be used in comparison with other numbers from the same tool. Here, fragmented desktops scored a 4763.2, while those which leverage the services of external defragmentation solutions scored a 5484.6. Thus, the net gain in overall system performance in this single test was around 14%.

- *Increased time to power on.* It can be argued that one of the most resource-intensive activities on any desktop or laptop system is related to its powering on. The bootstrapping as well as shell and user interface-instantiation processes require the involvement of numerous system components, all of which must occur in a very short period of time. Similar testing using Microsoft's Xperf.exe tool has shown that a fully-optimized disk drive can improve power on performance by an average of 3 to 5 seconds. Although this may not be dramatic for desktop users, this time savings is a boon for laptop users. This improvement in performance also extends to the hibernation process, whereby a laptop is put to sleep and later revived without requiring a full power-on process. As this process requires the creation and maintenance of a large hibernation file, its fragmentation further increases the process to revive a sleeping laptop.
- *System crashes and freezes.* As discussed in the first article, the process of fragmentation quickly spreads individual pieces of data into multiple locations. This widespread shattering of individual data files increases the chance that their later reassembly may fail, or may force a system freeze during the reassembly process. Eliminating fragmentation on a file system removes this variable from file systems, ensuring that files can be gathered from disk in a contiguous fashion.
- *Performance impact to existing enterprise services.* Lastly, the impact of fragmentation has a dramatic effect on other enterprise services, notably those that have a high reliance on disk and file system resources. Consider common business services such as antivirus and anti-malware. The mission of these agent-based solutions is to monitor the file system and processing for the potential intrusion of malicious code onto the system. Both real time and scheduled scans are often required for full assurance, so their processing is directly affected by the performance loss associated with data fragmentation.

The impacts on individual desktops and laptops are important to ensuring high levels of IT customer satisfaction. Yet the role of defragmentation doesn't stop at the data center's doorway. Inside that data center are another set of Windows OSs that operate in a server role. They too are impacted by the performance loss associated with file fragmentation, although any performance reductions here are experienced by a much larger audience than with any single desktop or laptop. Consider their additional situations:

- *Decrease in overall performance, particularly with very large files.* Implementing a proactive approach ensures that files make their way to disk in a non-fragmented state, and there is little to no need for later defragmentation to occur. Reactive defragmentation can be affective to resources on servers with very large storage requirements. It is particularly resource intensive when files are exceptionally large, such as those used by virtual machines or databases. Leveraging a defragmentation solution that uses the proactive approach means eliminating this performance impact to your servers.

- *Reduction in backup performance and increase in backup windows.* Files and folders must be reassembled before they can make their way to tape. Thus, the incremental process of archiving copies of your servers' data can take dramatically longer when not properly optimized for performance. This delay is further problematic as it increases the window of time required to complete backups, potentially complicating other off-hour tasks required in the data center.
- *Reduced ability to undelete files.* When a file is fragmented into multiple pieces, that file is spread across the disk's area. In cases where files are accidentally or maliciously deleted and require un-deletion, such a fragmented file has a dramatically lower chance of a successful restore. This happens because its individual pieces have a much greater likelihood of being overwritten by other data after the deletion event. This chance grows as the amount of time between the deletion event passes, giving the file system more opportunities to overwrite pieces of the original file.
- *Dramatically reduced performance of virtual machines.* With their entire disk subsystem consolidated into single files on another server's disk, the processing of virtual machines is exceptionally dependant on file system performance. When the very large disk files associated with virtual machines grow fragmented—a situation that is particularly problematic when virtual machine disk files are configured to grow as needed—the resulting reduction in the virtual machine's performance can be dramatic. This is the case for both the virtual machine's file on its host disk as well as fragmentation within the virtual machine's disk drives.

Defragmentation Equals Performance

As you can see through the examples discussed in this article, defragmentation is indeed primarily about your systems' performance. By implementing a policy of defragmentation that corresponds to established best practices and modern approaches, you will ensure the highest levels of performance for the systems in your network. This makes users happy while reducing the need for costly and unnecessary hardware upgrades.

Yet, throughout all this discussion, the question is begged *Doesn't Windows Have This?* A not-inappropriate question, the Windows OS does indeed arrive with its own built-in defragmentation solution. The third and final article in this series will discuss compelling characteristics of that native solution in relation to the capabilities your business needs.