Defrag Tech Spotlight

The End of Defragmentation?

Do New Storage Technologies (SAN/RAID/SATA) Make Defragmentation Obsolete?

Preface:

The definitions and explanations (of technologies) provided throughout this report focus on their specific use in the paper. There are new advancements and variations of these technologies beyond the scope of this paper's purpose and therefore, are not covered. This paper is also focused on Microsoft Windows and the Windows file systems.

Overview:

So, do new storage technologies make defragmentation obsolete? The quick and dirty answer is simply: No. File defragmentation is still a vital solution for peak system performance and reliability.

Due to the significant complexity and breadth of the software and hardware used in modern storage environments, from disk-level technologies to massively scaleable network storage facilities, there are many myths and misconceptions regarding the continuing need for disk defragmentation. Although it is understandably easy to accept many of them as replacements for file/disk defragmentation as many seek to solve the same issue, the fact remains that *the disk is the weak link*.

The purpose of this paper is to define the technologies, how they contribute to I/O throughput, and how the various new solutions can work *together* with defragmentation for optimal disk subsystem performance.

> *Regular jobs to defragment your disks will have a positive impact on performance. -HP*

If there is one key piece of data to remember after reading this document, it is that a disk file system is abstracted[1] from the actual underlying hardware and software that make up the storage subsystem. In other words, those same underlying disk systems (software and hardware) have no knowledge of what the file system is doing. The first section of this paper will follow I/O from start to finish through some of the various layers of abstraction.

A Q&A section will then present the various advanced technologies in detail, and how they relate (or do not relate) to file fragmentation. The Q&A section will also provide recommendations on best practices for applying file defragmentation with the respective hardware and software discussed.

[1] In the context used here it means: separated from other real or virtual components.

Breaking Down the I/O Path:

Single Disk Environment:

With any non-cached disk I/O[2] activity there is always a "block" involved. A block (also known as a sector) is the smallest unit in which data is transferred to and from a disk device. A block is created by the disk drive manufacturer in a *low-level format*. It is a physical location, of a set size (typically 512 bytes), which has a unique address from any other block on the disk. No matter what other technologies are layered on top of the disk to access data, the block always exists as the smallest indivisible unit.

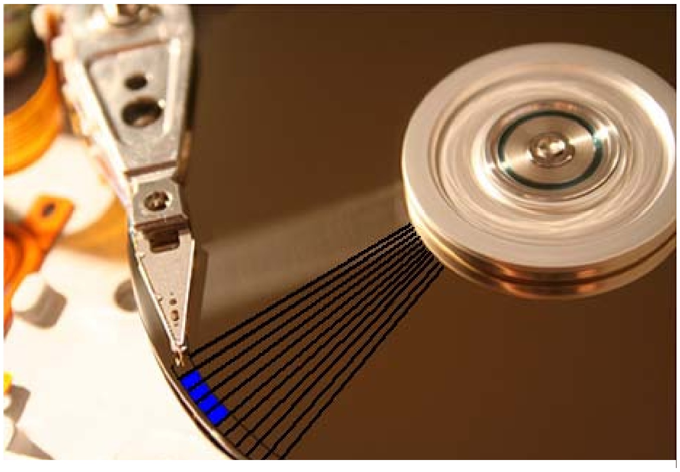


Figure1 – Sectors on a disk formed into a 2KB cluster

For a disk device to connect into the computer's system bus, it must use a host bus adaptor (HBA). That HBA is often built in to the motherboard for SATA and EIDE based disks. The HBA is hardware that extends (i.e., the adaptor part of the name) the controlling computer's (the host) circuitry (the bus) to other, typically storage, devices. The use of an HBA requires a software driver be loaded into the operating system.

The disk controller describes the firmware that controls a disk drive. It interprets a Logical Block Address (LBA) to locate data on a "block" (sector) level. Disk controllers require that a software device driver be loaded into the operating system to support two-way communications in the form of I/O Request Packets[3] (IRP).

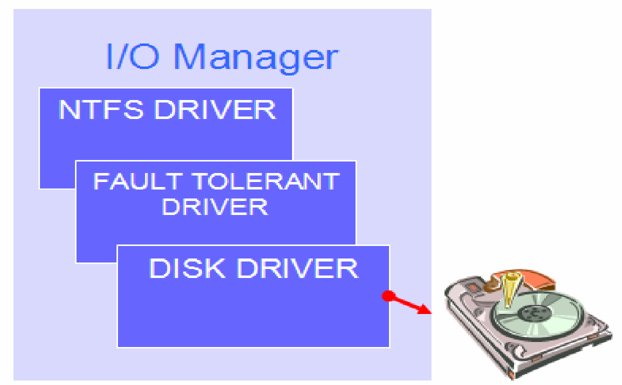


Figure1.1 – I/O path from OS to physical disk

The file system, with respect to disks, is a high-level format mapping of logical units known as clusters. The file system uses an index to assign clusters to file objects (e.g., report.doc). In NTFS this data is stored in an index file called the Master File Table. Clusters are mapped to files by recording the logical cluster numbers, on which the file is stored, in the index record for the given file[4]. A good analogy would be the index in the back of a book, directing you to the page number where a keyword is used.

A file system, such as NTFS, operates in Kernel mode[5] in the operating system. For each "logical" fragment in the file system, a separate disk I/O must be generated and passed on to

[2] I/O (input/Output): refers to the data transferred from one device to another.

[3] Kernel mode communication structure between device drivers and the operating system.

[4] For a detailed explanation, see *How NTFS Reads a File* in the Reference section at the end of this paper.

[5] The kernel defines the trusted core system component responsible for managing hardware operation requests (e.g.,process time, disk and memory management). To run in kernel "mode" defines the execution of instructions at this level (ring 0).

the disk subsystem. Disk subsystems, no matter how intelligent the controller, operate at the block level and cannot recognize a file object. Therefore they cannot re-assemble or pool incoming I/O requests related to logical fragments and minimize the amount of physical motion.

Multiple Disk environments:

I/O in disk subsystems wherein data striping is used follows the same path as the single disk, but the data, and the I/O load, are physically distributed over many disks.

Disk arrays are commonly used with server systems. A disk array is a physical storage container with power supply that contains multiple hard disks, a disk controller card for the array, with a cache, and as a standard, offers disk striping and fault tolerance. They connect in to the host operating system via a host bus adaptor (HBA). Originally the term LUN (Logical Unit Number) only meant the SCSI disk address on an array for a particular disk, but it is now commonly used to represent the physical disk array when it is implemented in a Storage Area Network (SAN) as a logical volume.

Here is the I/O path:

I/O Request:

↓ Application requests to read a file	(User Mode)
---	-----------------
↓ The request is passed to File System	(Kernel mode)
↓ The File System maps the file clusters to an LBA and passes it to the driver (HBA)	
↓ HBA driver maps LBA to particular physical disk in array	
↓ Physical disk onboard controller maps the request to a specific block	

I/O Retrieval:

↓ Physical disk acquires specific blocks	
↓ Disk array controller acquires blocks from disk	
↓ Blocks are mapped to LBA’s and passed to the file system	
↓ File system maps LBA to file clusters and passes to application	(Kernel mode)
---	-----------------
↓ Application receives file	(User Mode)

The request then traverses this path back up, in sequence without skipping a step, to the application.

> *"I think defrag is an excellent tool for keeping your performance and the health of your drive up to par. So the larger these drives get the more data people are storing on them, and the more you store on them, you edit and save and pull it back up another day. It sort of gets spread out across the hard drive… so when you defrag you are pulling all of these files closer together. … he doesn't have to search over this 750G drive to find pieces of a file, they're all aligned…."*
>
> *- Joni Clark, Product Manager, Seagate*
> *(as heard on the Computer Outlook Radio Show)*

Questions and Answers Section:

Question: “I have a top of the line disk in my new workstation (SATA/SCSI), so do I still need to defragment?”

Answer: Yes. To fully answer the question, let’s break down the various technologies that make up modern hard drives, and address them one at a time.

A. Queuing and Seek Optimization:

There are a number of disk level algorithms to minimize the impact of physical limitations such as rotational latency (waiting for the disk to spin back around). They include variants of elevator-seeking and shortest-seek-first. These algorithms leverage the disk buffer, prioritizing retrieval of data physically closest, by measure of the data’s cylindrical location and/or how close the requested data is to the current location of the disk head[6].
One of the next considerations that might come to mind is doesn’t “disk queuing” eliminate the need to defrag?

Native Command Queuing (NCQ) is a technology that allows a SATA drive to re-prioritize and queue disk requests while completing others. It’s kind of like multi-tasking at the disk level. The big benefit is that it excludes the CPU from having to be active in backlogs from what is the slowest component in the modern PC. SCSI disks have led the way, and continue to do so, supporting up to 256 queued commands.

The answer to whether seek optimization and disk queuing eliminate the need to defragment, the answer is simply no. While seek algorithms improve on rote data retrieval methods of the past, they cannot account for fragmentation as they are “block” based. They will organize and prioritize data retrieval based on physical location of data blocks, not per file object. Queuing will improve on prioritization strategies and improve overall seek time for asynchronous I/O, they also do not address fragmentation, as they too are block based and do not optimize the activity for a particular file. An important point to make is that almost all I/O is synchronous, so unless the system is specifically addressing numerous simultaneous users, the effective applied benefit of queuing is minimal.

B. Disk Cache:

Caches are commonly available on the disk/disk array controller. This is a volatile memory , requiring constant power, through which data is stored temporarily (buffered en route to being written to the disk (*write-back*).

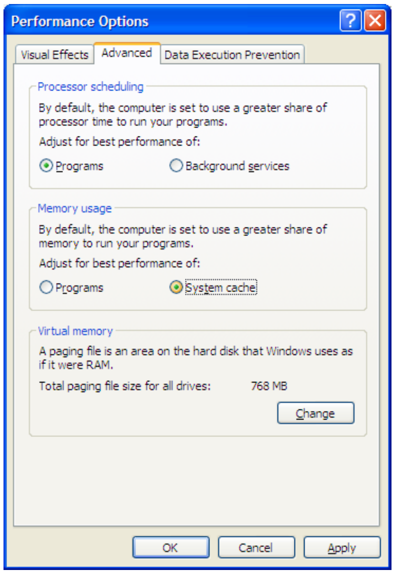
The cache can be of benefit for re-reading data that has been loaded into the cache (either from a recent read or write), as the cache duplicates data found on the disk platter. Reading from cache improves performance as it eliminates the need to retrieve data from the disk platter.

Many controllers also offer *read-ahead* (pre-fetch) caching for sequential I/O. This attempts to pre-read blocks from the disk and place them into the non-volatile storage, ahead of the

[6] For more information on disk architecture see *The Shortcut Guide to Managing Disk Fragmentation* in the reference section.

actual system request. Data located in physically non-contiguous blocks (i.e., due to file fragmentation) impedes read-ahead technologies, since disk devices do not map blocks to file objects. As a result, they do not read in the proper data. For this reason, file defragmentation will aid controller read-ahead efforts.

The primary purpose for this on-drive cache is sequential read improvement, it does not offer significant benefit to random reads.



Processor scheduling

By default, the computer is set to use a greater share of processor time to run your programs.

Adjust for best performance of:

● Programs ○ Background services

Memory usage

By default, the computer is set to use a greater share of memory to run your programs.

Adjust for best performance of:

○ Programs ● System cache

Virtual memory

A paging file is an area on the hard disk that Windows uses as if it were RAM.

Total paging file size for all drives: 768 MB

Figure1.2 – System Cache on Windows Server

It should also be noted that the operating system maintains a system cache as well. This cache is housed in the computer's system RAM and allows an operating system to consolidate I/Os in a buffer (write-back cache), making for more efficient I/O. Recently accessed data can be retrieved from this cache, without need to request it from the disk.

C. Electromechanical:

Serial ATA (SATA) drives are the new standard for desktop PCs and most laptops, while SCSI has been the mainstay on servers for over a decade, and Fibre Channel (FC) becoming more common in SAN environments.

Today's fastest disks spin at 15,000 RPM with average seek times on these high-end drives measured in the mid 3 milliseconds. While this is a great improvement over the 5200 RPM disks of the 1990's, it's a far cry from the operational speed of CPU and memory, which have improved at a far greater pace and are now measured in nanoseconds.

High end SATA, serial attached SCSI (SAS) and Fibre Channel attached disks transfer rates move data through at a rate of 300MB/sec – 400MB/sec. Given that CPUs and memory can process 14 GB/sec+ of data (processor-to-system bandwidth), disks simply cannot keep up.

Keep in mind that the reported transfer rates already include all the technologies that optimize data access mentioned above. Some disk models are faster than others, and some interfaces allow greater throughput than others. The performance trade-off is typically price. Faster drives usually mean higher costs.

Summary:

Queuing and disk seek algorithms do afford improvements for inherent mechanical performance restrictions, but they simply do not make up for the fact that these devices cannot keep up with electronic speeds. If all data could be permanently maintained in high-speed RAM, fragmentation would not be the performance issue it is, but price-per-GB of RAM isn't affordable or appropriate for long-term or mass storage. It should also be noted that huge caches still do not help with file writes, which suffer in a fragmented free space environment.

When a "bottleneck" in a process (any process) is restricted, the entire process is significantly impeded. The greatest gain to be realized is the resolution of that bottleneck. Arguments that "today's disks are faster" and hence fragmentation is not an issue, hold no validity. Based on transfer rates alone, one can quickly conclude that the disk drive is still the weak link in the chain.

Recommendation: Automatically defragment SATA/SCSI/EIDE(IDE) disks in *real-time*.

Question: "I have a hybrid disk in a new laptop and a solid state disk (SSD) in another new laptop, so do I still need to defragment?"

Answer: Yes.

Hybrid Hard Drives (HHD) offer a non-volatile RAM (NVRAM), in typically larger capacities than formerly available with on disk caches. The added benefit is that the added memory capacity does not require a constant power source.

Hybrid drives can improve data access by essentially caching data on the attached NVRAM. However, the NVRAM is limited in size and durability, and data prepended to the NVRAM relies on a predictive algorithm that does well for commonly used data, but cannot account for all activity. Disk platter access is still necessary, even on only moderately active systems. Spinning up a "resting" disk will extend total seek time, more than disks that are not spun-down.

Solid State and similar NAND Flash disks do not suffer from electromechanical read latency as do rotating magnetic media. While asynchronous read operations are efficient, SSD/Flash is not the best choice for sequential I/O or heavy write activating due to erase-on-write requirements. Fragmented free space will dramatically impact write speed on these devices.

Recommendation: Defragment Hybrid drives on an occasional basis (e.g., once a day or once a week). Make sure to use an advanced tool, such as Diskeeper, that focuses on performance improvement, thereby not causing excess write activity to the NVRAM to achieve a "pretty disk display". Defragment SSD disks on an infrequent basis, once a week or once a month, with a focus on thorough free space consolidation (e.g., a tool such as Diskeeper).

Question: "I have a top of the line RAID array in my new server, so do I still need to defragment?"

Answer: Yes

Fragmentation degrades the performance of a RAID environment due to the unnecessary generation of I/Os issued by the file system. All these numerous and separate I/Os are then passed to the RAID controller to process (read or write). The RAID controller, unaware that the multiple I/Os all map to the same file, treats each I/O as a separate entity. A file object split into multiple I/Os is more likely to be interspersed with other disk I/O in a RAID stripe, than if the file I/O delivered to the controller was single.

> *"Just as with a single disk, files stored on RAID arrays can become fragmented, resulting in longer seek times during I/O operations."*
> *– Microsoft*

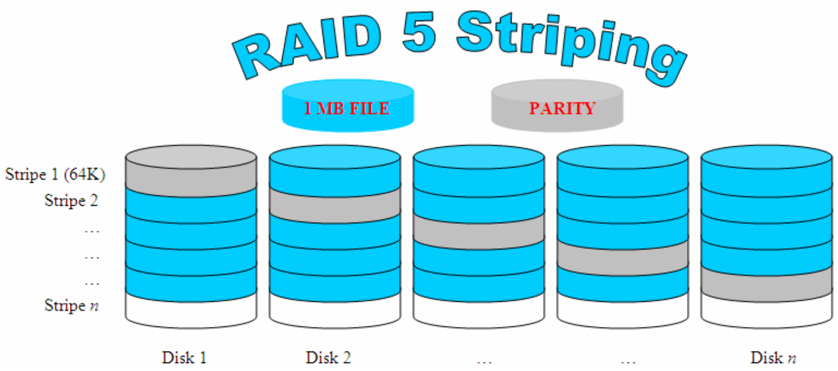


Figure1.3 – 1MB file in one logical I/O, evenly striped (w/parity) across 5 disks

Defragmenting files at the file system level and consolidating data into a single I/O can, depending on the RAID controller, better fill the entire (e.g., 64K) chunk (RAID stripe) size with that I/O; now taking full advantage of the RAID.

If a logically fragmented file does get interspersed with other I/O (due to the fact that multiple I/Os have to be generated to write a file), it is theoretically possible that the data for that file is not evenly spread across the disks. Following that theory, the possibility of uneven file writes is increased on busier disks using smaller stripe/chunk sizes.

Figures 1.3 and 1.4 depict what can occur. This is for illustrative purposes only, as each stripe can, of course, contain data from multiple files all combined into a single stripe.

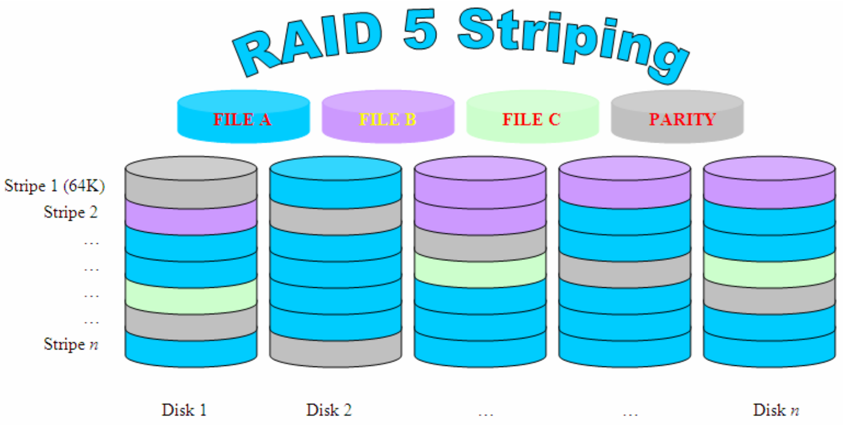


Figure1.4 – 1MB “File A” unevenly written as simultaneous I/O writes are coalesced

Re-writing of parity data is possible with disk defragmentation. For that reason, it is important to have I/O-sensitive defragmentation technology. This will prevent I/O overhead at the controller cache.

The type of distribution methodology also makes an impact in the layout of data. Proprietary metadata in the form of a tree structure may be used to map out data within a storage subsystem. In other cases simple correlations are created. Where vendors claim that there is no “fragmentation” this refers to the metadata (or other methodology) used for file layout within the storage system. This does not refer to fragmentation at the disk file system level (NTFS).

Advanced/intelligent controllers and caching:

In a RAID device, read-ahead and write-back caching is also done at a block (not file) level. Read-ahead is valuable for sequential reads, and advanced technologies apply read-ahead in an intelligent manner. As was noted earlier, the issue is that if, at a block level, the file object requested by the operating system is not contiguous, read-ahead caching will not operate as well as it could if file fragmentation was handled.

Physical members [disks] in the RAID environment are not read or written to directly by an application. Even the Windows file system sees it as one single "logical" drive. This logical drive has (LCN) logical cluster numbering just like any other volume supported under Windows. … fragmentation on this logical drive will have a substantial negative performance effect.

- Diskeeper Corporation

Write-coalescing describes technology used by the RAID software's controller cache to buffer large sequential incoming data, (e.g., in a FIFO method), until an entire stripe's worth of information is gathered. The buffer also supports the generation of parity data prior to writing, in order to avoid a mechanical penalty for writing parity on-the-fly. The buffer is; of course, block based, waiting for enough I/O write data before it stripes the data (and parity) across the disks. It does not natively coalesce data from a single given file object. Defragmented files and free space can improve the performance and viability of write coalescing, by increasing the likelihood of sequential I/O writes.

Adjusting Queue Depth:

While disk level queue-depth mitigates taxation of the CPU to some degree, it still bottlenecks the disk itself, so one must be careful when modifying this function. Done correctly it can increase performance, but done improperly it can be damaging and impact available resources for other devices on that same HBA, decreasing the throughput for those devices. As an example, it is highly recommended to monitor and adjust SCSI queue depth in virtual machine environments to accommodate the increased disk activity that multiple VMs generate. Setting the value too high, however, can expose data in queue buffers to corruption, SCSI time-outs, and cache flushes.

…If an application has to issue multiple "unnecessary" I/O requests, as in the case of fragmentation, not only is the processor kept busier than needed, but once the I/O request has been issued, the RAID hardware/software must process it and determine which physical member to direct the I/O request.

- Diskeeper Corporation

Summary:

For defragmentation to be of benefit, it is a misnomer that the end result must provide a one-to-one logical-to-physical correlation. In the case of fault tolerant or I/O distribution efforts of RAID or virtualization, it is standard for the data to be physically split across multiple disks. Defragmentation never forces a one-to-one mapping, nor does it need to in order to provide benefit. However, a single I/O delivered to the RAID controller, for a given file, is more likely to be optimally striped across the RAID disks than multiple I/Os to a file interspersed with other file I/O.

Recommendation: Automatically defragment RAID arrays in *real-time*.

Question: "I have a new Windows based NAS box that I just plugged into the network, so do I still need to defragment?"

Answer: Yes

A NAS box is essentially a disk array, with an operating system (Microsoft offers the Windows Server Appliance Kit and Windows Storage Server/Windows Unified Data Storage Server through NAS OEM manufacturers) that can be plugged into the network. It is a plug-and-play file server.

Recommendation: Automatically defragment NAS devices (running Windows) in *real-time*.

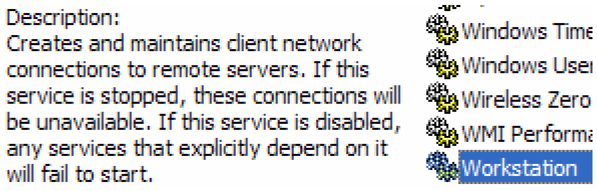
Question: “I moved all the network data to a new top-of-the-line SAN with Storage Virtualization, so do I still need to defragment?”

Answer: Yes

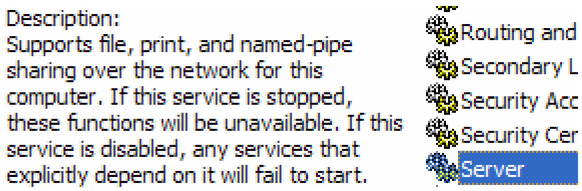
The purpose of a SAN affords the administrator the ability to make remote disk arrays appear to be local. It does not matter how they are connected, iSCSI, Fibre, etc…

There are a great many tangential implementations in SAN technology so this section will focus on what is the standard application or usage of the typical SAN technologies. This does not, by any means indicate that a new, or proprietary technology eliminates the fact that fragmentation impacts performance, it is simply that it is unnecessary to expound on those details for the purposes of the paper.

The statement that a SAN volume appears local to the Windows operating system is an important concept. Windows does support “remote file systems” just as it supports “local file systems” (e.g., NTFS.sys). In Windows, the remote file system includes a client (requestor) and a server (provider) component. A part of this remote file system are the Windows services “Workstation” and “Server”.



Description:
Creates and maintains client network connections to remote servers. If this service is stopped, these connections will be unavailable. If this service is disabled, any services that explicitly depend on it will fail to start.



Description:
Supports file, print, and named-pipe sharing over the network for this computer. If this service is stopped, these functions will be unavailable. If this service is disabled, any services that explicitly depend on it will fail to start.

This remote file system (also called distributed file system) is the mechanism used when, as an example, connecting over Ethernet to mapped shares on a file server. The protocol used to communicate between the requestor and the provider in a Windows network is known as Common Internet File System (CIFS), which is a Microsoft variant of IBM’s Server Message Block. CIFS has many other Windows network uses, and while improving, some of its limitations restrict usage in environments where remote data performance is vital. Hence, it is one of the driving forces behind the creation of other technologies such as SAN and NAS.

I’ve implemented storage virtualization, so do I still need to defragment my local file system?

Storage Virtualization is commonly used in SANs. This technology essentially abstracts “logical storage” (what the operating system sees and uses – i.e., the file system) from physical storage (the striped RAID sets). The key differentiator in virtual storage is that the multiple physical storage devices (e.g., a RAID array) are combined into one large grouping, on top of which a virtual storage container is created.

Perform regular defragmentation of the file system to ensure optimal performance
-EMC

SAN file systems (a.k.a. cluster file systems) such as VMFS from VMware or EMC's Celerra, are a third and different category of file system known as shared-disk file systems and are the backbone of storage virtualization (different from previously defined Local or Remote file systems). An operating system defragmenter, such as Diskeeper, only recognizes the "local" disk file systems that it natively supports. Vendors of proprietary files systems typically include specialized technologies to optimize performance. These file systems are the foundation for storage virtualization.

I/O Mapping and Redirection

Storage virtualization uses metadata to properly channel I/O. Software on a storage virtualization device (such as a SAN Switch) will translate logical disk locations to physical disk ones.

Here is an example:

1. A storage virtualization device gets a request for a logical location of LUN#1, LBA 32
2. It then performs a metadata lookup for that address and finds it actually maps to LUN#4, LBA16[7].
3. The device then redirects the request to the actual physical location of the data
4. Once it retrieves the data, it passes it back to the originator without the originating requestor ever knowing that the request was completed from a different location than what it knew.

The fact that there is not a one-to-one mapping of file system clusters to LBAs (due to LUN virtualization) is not an issue. Logical, file system level fragmentation causes the operating system to generate additional I/O requests to the virtualization software. Using metadata, the software then redirects I/O from the logical disk to its physical location.

The local disk file system (e.g., NTFS) does not know of, nor control the physical distribution or location in a virtualized storage environment, and as a result of fragmentation, NTFS has to make multiple requests regardless of the physical or virtualized storage environment.

> *The SAS (Serial Attached SCSI) software does not know what the total size of a file will be when it is created; therefore, it can not be contiguously allocated. File fragmentation is a problem because additional file system activity must take place to access a file that is stored in multiple, noncontiguous locations on a volume. When defragmenting a volume, all the files on the volume are rearranged so that each file is in one contiguous extent.*
>
> *-HP*

In SAN file systems, block size (the smallest addressable virtual unit) is a configurable metric and varies based on the software used. Vmware's VMFS, for example supports 1MB to 8MB blocks.

Logical Cluster Numbers (LCNs) are a file system construct used to map a file in an index table (e.g., Master File Table in NTFS) to LBAs. Disk Controllers take those logical blocks and make the appropriate translation to a physical location. Disk controllers do

[7] With disk arrays often abstracting LUNs out of large RAID sets or RAID Volumes, multiple LUNs can be presented from a single stripe set and presented to different hosts.

not—no matter how “smart” they are—independently map fragmented file I/O into consecutive or linear block requests. They cannot “pool” incoming block-based data back into a file.

This means that regardless of the fact that the file system does not map directly to a physical location, file system fragmentation will create the exact same kind of phenomenon on RAID as it does on virtualized storage (multiple RAID arrays group together).

SANs can offer extremely efficient and high-performing data storage, but it is not the job, nor within the scope of ability for a SAN system (hardware or software) to address file system level fragmentation. Proprietary technologies employed by one vendor can be more efficient at retrieving data blocks than another. Architectures can vary as well. No matter how efficient data retrieval can be, and how much physical disk limitations can be mitigated, the overhead on the operating system that is retrieving the file is beyond the scope of SAN technology and is impacted by file fragmentation.

So, to answer the question, yes local disk file defragmentation is still necessary.

Does Defrag cause an issue with Thin Provisioning?

> *We use it [Diskeeper] on our big SQL box (8 way processor, hundreds of gigs of space on a SAN, 16 gigs of RAM) and it has increased our disk performance by a factor of about 8 or 9. We were looking at adding more spindles to our SAN to help with some disk I/O issues we had, but this wonderful software did it for us.*
> *- Dave Underwood, Senior Engineer, CustomScoop*

Given that allocated disk space often goes unused, a technology called Thin Provisioning was developed to make it appear more disk space existed *virtually* so SAN storage can allocate physical space dynamically to the volumes that need it. Space in the SAN file system is allocated on an as-needed and amount-needed basis in a pool shared by multiple servers. Provisioning accommodates the unpredictability and allocation of future storage growth needs and eliminates the need to assign storage to one volume/computer when the system is built.

In a technical article from leading SAN vendor Compellent called “Dealing with Thin Provisioning and High Water Marks” they discuss various operating systems tendencies to either reuse deleted blocks or write to previously used blocks. A high water mark is the term that defines the last written block of data. That high water mark always increases and never decreases (on Windows), indicating less available space to the SAN. This creates a problem in properly provisioning space.

In tests done by Compellent engineers, they noted that when Diskeeper defragmented a thin provisioned volume, deleted space was returned as available space to the SAN, preventing over-allocation of the SAN file system, thus supporting proper function and use of only the space that is needed. Due to functional differences, the native Windows Server defragmenter did not return the same positive results.

If you implement thin provisioning, it is recommended to check with both your SAN technologist *and* defragmentation software vendor, for proper configuration.

Can proprietary Device Specific Modules (DSMs) solve file fragmentation?

No. This is a common misconception, even by many SAN vendors' knowledgeable technical support staff. A SAN vendor may create a DSM for fault tolerance or performance purposes. Typically an I/O request travels one path (as described earlier in this paper) from application to physical storage location. DSM allows a vendor to design alternative "paths" to the physical data storage, in the event a component along the path breaks (e.g., a bad cable), or bottlenecks under heavy load. In either of these events the DSM can re-direct the I/O down another pathway. This is called a multipath I/O driver (MPIO). It is great for optimizing the performance of block level requests generated by the file system, but cannot minimize the overhead that file system occurs in generating multiple I/Os for one file object. It must accept the multiple unnecessary I/Os and optimize the retrieval of those multiple requests as best as possible.

> *We've seen a huge disk performance gain…using Diskeeper to defragment all databases/images/documents stored by our clients, which include over 1.5 TB of SQL data and file storage data. Data is stored on an IBM FastT500 SAN array on three 10-drive nodes (600GB, 600GB, 1.2TB respectively).*
>
> *-Doug Robb, VirMedice. LLC*

MPIO (disk class drivers) reside below NTFS.sys in the I/O stack, and are reliant on IRPs initially generated from the file system and passed down the stack. "Fixing" local disk fragmentation is not the job of the SAN vendor, nor even their responsibility, as the issue occurs are a higher level (closer to the requesting application) than a SAN is, or should be, integrated with system I/O.

A quick overview of file requests through the Windows storage stack[8]:

↓ Application (e.g., SQL)

↓ I/O Manager

↓ NTFS.sys

↓ Volsnap.sys

↓ Disk.sys (e.g., SAN replacement of this driver such as MPIO)

↓ Hardware

While SANs implementing DSM can distribute I/Os more efficiently than Direct Attached Storage (DAS), fragmentation will still create application slows, as the operating system where the requesting application resides still has to generate more I/Os than it should.

Do resource-sensitive technologies to throttle disk activity work in a multi-headed SAN?

One of the technologies offered by defragmenters is I/O throttling. The concept here is to monitor I/O traffic, and throttle application related disk activity (i.e., defrag) until the I/O pipeline was free. Collection of I/O queue data is dependant on counters returned from the operating system on which the application is installed. In cases where multiple operating systems (multi-headed) connect to common shared disks, as in a SAN, I/O throttling techniques will not function appropriately. The case being that an application using I/O throttling may detect that a shared disk array is not busy, but a secondary server also using that same array may be processing a very disk intensive operation. In that event disk

[8] Details such as FAST I/O, cached-data and other intermediate drivers, but the relevancy is insignificant for this discussion.

contention would occur, bottlenecking the disk intensive process originating from the second server.

Diskeeper Corporation’s proprietary InvisiTasking™ technology (which eliminates overhead on direct attached storage) will provide more effective resource-sensitivity for storage networks through granularity of its actions. However, with the possibility of overhead conflict in more I/O demanding SANs, Diskeeper Corporation recommends proper evaluation of the environment to determine if Diskeeper defragmentation time frames are better suited to off-peak production hours.

Can defragmentation of my LUNs move important data onto slower sectors of the RAID set?

Yes, with simple defragmenters this can occur, and is why it is important to use an advanced technology [Diskeeper’s I-FAAST™] that measures data transfer performance and applies this information to properly sequence frequently used data. For that matter many actions can cause data to move to slower segments as well. Opening, modifying, and saving office documents causes a re-write of the new file to a new, potentially less ideal, location.

For that reason, it is also important to implement proper partitioning strategies when setting up a SAN.

Recommendation: Automatically defragment SANs. If I/O conflicts on the disk arrays or network/channel bandwidth competition are noted, relegate automatic disk defragmentation to off-peak hours.

Summary:

The point is that new disk subsystem technologies absolutely aid in a better performing system and are important considerations, but they do not solve issues generated at the file system level such as file fragmentation.

Physical storage devices and controllers will optimize the location of blocks across the underlying physical spindles according to their proprietary methods, but none are involved with how the file system requests I/Os. The need to defragment SAN, RAID, SATA, SCSI, NAS, HHD, SSDs devices continues today, just as it has in the past.

When bottlenecks occur in the disk subsystem, file fragmentation is a factor that should always be investigated as a contributing factor. To gauge the impact of fragmentation, use performance monitoring tools such as PerfMon, Iometer, or hIOmon. The appendix at the end of this paper provides examples and links to these tools.

> *We use Diskeeper EnterpriseServer on our main file server. This particular system has about 4-5 TB assigned to it from our SAN and needs constant defragmenting due to heavy use.*
> *-Martin Gourdeau, Network Admin, Electronic Arts*

For all the technical data provided on why fragmentation is still relevant with new storage technologies, other real world factors make evaluating fragmentation a worthwhile cause.

Primarily, investigating fragmentation is inexpensive; requiring only some investigatory time. The evaluation software can be obtained for free, and the time required to test is far less than that involved in evaluating hardware solutions. Licensing and managing defragmentation software will be far less expense than hardware solutions and will likely provide a significant and immediate return on investment.

Appendix A

Gauging the impact of fragmentation:

PerfMon:

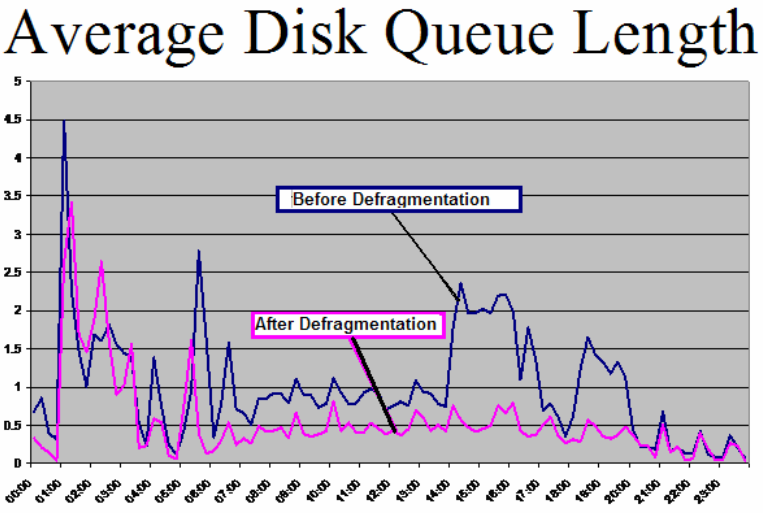
To determine fragmentation’s impact on a disk subsystem (single disk or RAID), you can employ performance monitoring technologies. Windows includes a built-in tool called PerfMon that can collect and graph this data. Specifically, you will want to direct it to the *PhysicalDisk* object. Performance monitoring for purposes of determining event-based changes (such as defragmentation) requires proper before (baseline) and after comparisons. This means that a similar extended period (e.g., one week) must be compared to determine improvement. No other changes, such as adding new hardware, can be introduced during the test periods. The periods measured must cover, to the degree possible, the same work load.

Here is a sample scenario:

1. On a Friday afternoon, install, but do not activate, an enterprise-class disk defragmenter [Diskeeper Server], and run the tool’s native analysis functions.
2. Save the defragmenter’s analysis reports.
3. Start the PerfMon baseline on a Monday and let it run without any other hardware/system settings changes for one full week.

    - Avg. Disk Queue Length (should have no more than 2 per spindle)
        - Avg. Disk Read Queue Length (used to further define disk queues)
        - Avg. Disk Write Queue Length (used to further define disk queues)
    - Avg. Disk Transfer/sec (should be less than 50-55 per spindle)
        - Avg. Disk Read/sec (used to further define transfer rate)
        - Avg. Disk Write/sec (used to further define transfer rate)
    - Split IO/sec (should be less than 10% of Disk transfers/sec value)
    - % Disk Time (should ideally be less than 55%, over 70% is typically an issue)
        - % Idle Time (to check legitimacy of % Disk Time)

4. Using the disk defragmentation software, run another analysis and save the results.
5. Activate the defragmentation tool the following Monday morning and let it run for two weeks.
6. Using the disk defragmentation software, run the final “after” analysis and save the results.
7. Compare (see figure at right) the first and last week periods and note changes (improvements) in the measured counters from week one (no defrag), to week three (defrag complete and still active). The disk defragmenter’s reports will provide you data on



the changes to file fragmentation as part of this before-and-after comparison.

8. If desired, stop defrag operations for the fourth week, and continue to monitor disk performance through week 5, to note reversal of achieved performance gains. Accompany this with another disk defragmentation analysis and compare the results of that analysis to data collected from week 3.

The handy Performance Monitor Wizard, available at Microsoft's website can ease the learning curve in setting up and using PerfMon.

No counter will independently determine the impact of fragmentation. If the disk is fragmented, many of these counters will show metrics higher than acceptable levels.

hIOmon™ by HyperI/O℠

Diskeeper partner HyperI/O has developed a full "file I/O performance" evaluation kit, targeted specifically at determining the impact of fragmentation on production systems. Due to its robust feature set, this is Diskeeper Corporation's recommend product/method for experienced Server Administrators familiar with benchmarking and performance evaluations.

Iometer

An additional benchmarking tool is Iometer/Dynamo (distributed as binaries). It is an open source I/O subsystem measurement and characterization tool. Iometer/Dynamo can be used to benchmark test environments. The key to benchmarking fragmentation with this toolset is ensuring the test file is created in a fragmented state. This can be accomplished by fragmenting the free space on a test volume prior to use of this tool.

Appendix B

References:

VMware (KB-1014) - Windows Virtual Machine Blue Screens When I Use SAN LUNs

Microsoft (Disk Fragmentation impacts RAID) - Windows Server™ 2003: Improving Manageability and Performance in Hardware RAID and Storage Area Networks

HP - Configuring Windows Server 2003 with HP Integrity Servers with SAS

Diskeeper Corporation - How NTFS reads a file

Diskeeper Corporation – Is Real Time Defragmentation Needed in Today's Environment?

EMC – Backup Storage Solutions for CLARiiON and Symmetrix

SQL Knowledge - How to Monitor I/O Performance

Diskeeper Corporation – File Fragmentation: SAN/NAS/RAID

Compellent Dealing with Thin Provisioning and High Water Marks (only available to customers)

Iometer.org – Downloads and documentation

Hyper I/O - Fragmented File I/O Metrics

Bibliography:

RealTime Publishers - The Shortcut Guide to Managing Disk Fragmentation (Mike Danseglio)

Diskeeper Corporation – FRAGMENTATION: the Condition, the Cause, the CURE (Craig Jensen)

Microsoft Press - Microsoft Windows Internals (Mark Russinovich, David Solomon)

O'Reilly – Using SANs and NAS (W. Curtis Preston)
