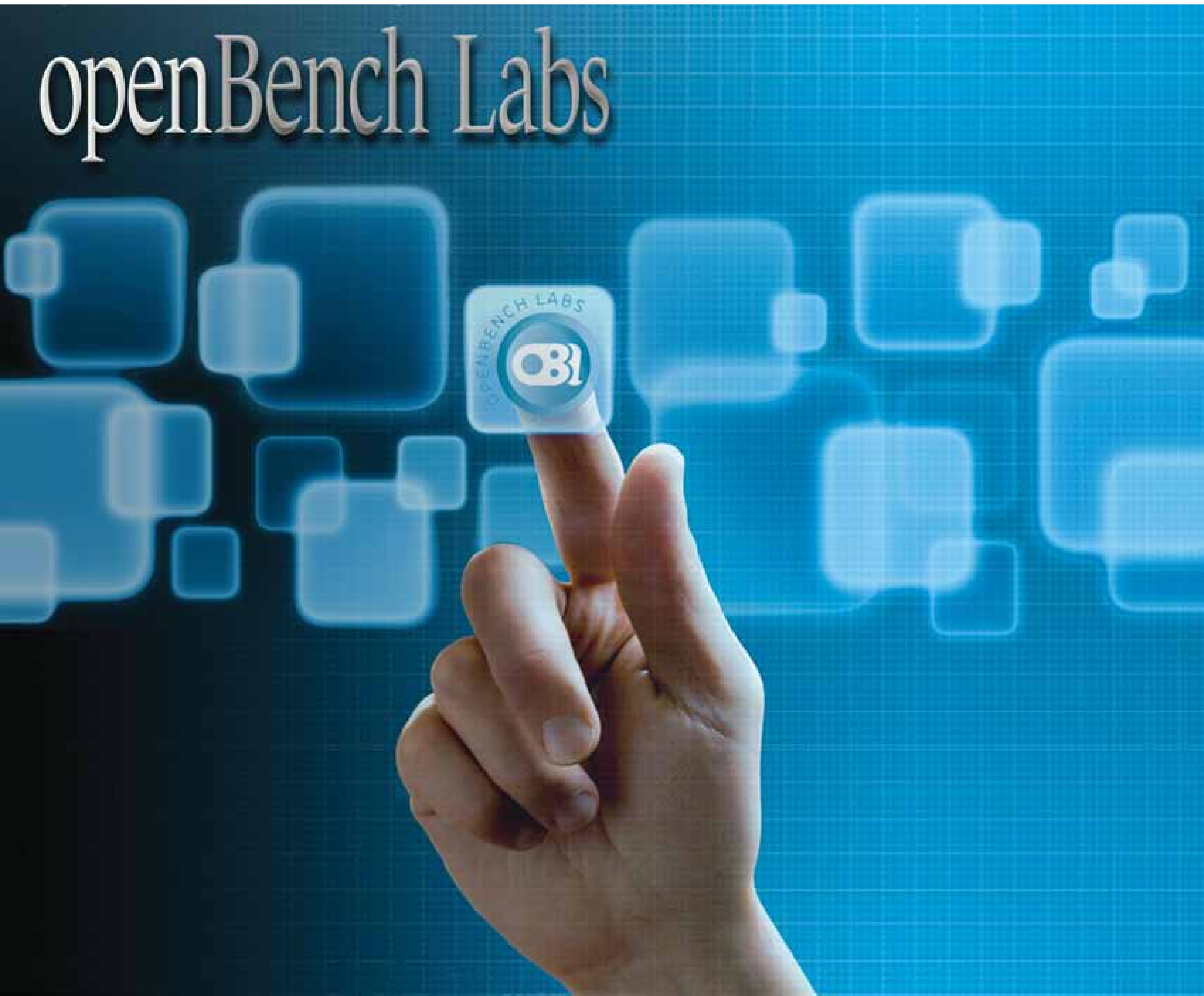
openBench Labs
Executive Briefing:

October 8, 2012

Condusiv Diskeeper 12 Intelligently Restructures Writes to Maximize Performance of Server and Desktop Systems

Improving Storage Efficiency by Increasing I/O Throughput and Decreasing I/O Operations



Executive Overview:

"In testing the effects of file fragmentation on the latest Windows server and desktop operating systems, Diskeeper 12 offered significant advantages for end users by maintaining optimal application performance with respect to both storage and CPU resources and equally significant advantages for IT operations by reducing physical I/O operations performed by storage resources, which reduces power consumption."

Why Read This Document?

If you are currently using the disk defragmentation utility bundled with a Microsoft® Windows operating system, third-party defragmentation software, or are concerned over the deleterious effects of defragmentation on a VMware® Virtual Infrastructure (VI), Diskeeper 12 by Condusiv® Technologies introduces an entirely new way to optimize storage performance and reduce many datacenter operating expenses associated with storage resources.

For this white paper, openBench Labs tested the ability of Diskeeper 12 to break the cycle of file fragmentation and defragmentation by preventing file fragmentation before it occurs, rather than remediation of file fragmentation after the fact. More importantly, openBench Labs measured the susceptibility of the latest Windows operating systems to file fragmentation and the effects of that fragmentation on operating performance and efficiency. All of the tests utilized by openBench Labs can easily be replicated by IT administrators, who will need to evaluate this product based on the answers to five key questions:

- **How susceptible are Windows 7 and Windows Server 2012 to file fragmentation?**
- **How does file fragmentation affect system performance and end-user perception?**
- **How does file fragmentation affect operating conditions on physical storage systems?**
- **How well does periodic defragmentation remediate the effects of file fragmentation?**
- **How well does Diskeeper 12 prevent file fragmentation?**

Findings & Benefits

In testing the effects of file fragmentation on the latest Windows server and desktop operating systems, Diskeeper 12 offered significant advantages for end users by maintaining optimal application performance with respect to both storage and CPU resources and equally significant advantages for IT operations by reducing physical I/O operations performed by storage resources, which reduces power consumption. Here are several key takeaway findings and their impact on IT operations:

Finding: Severe file space fragmentation occurred on Windows server and desktop systems through standard software installation and large file creation.

Following the installation and updating of software packages on a desktop system running Windows 7, new executable software files often exhibited several hundred file fragments. Greater file fragmentation occurred on a server that was running Windows Server 2012 and exporting virtual volumes as iSCSI targets. On the test server, iSCSI target volumes exhibited file fragments numbering in the thousands.

Finding: Windows Server 2012 and Windows 8 automatically schedule defragmentation processes for all logical disk volumes.

For the first time, Windows operating systems default to scheduling and running file defragmentation processes on all storage volumes on a weekly basis.

Finding: Default file defragmentation using the built-in utility failed to remove all file fragments and consolidate free space even when processes invoked multiple defragmentation and consolidation passes.

Following scheduled defragmentation processes with the built-in utility, heavy fragmentation of files and free space remained after scheduled defragmentation processes had completed and reported that fragmentation had been eliminated.

Finding: Under typical fragmented file conditions, a single process reading data sequentially exhibited a 50% reduction in throughput as the number of physical I/Os performed by the storage subsystem doubled.

Using Iometer in our standard workstation environment, which relied on scheduled defragmentation using the built-in utility, we could not create a contiguous test file. File space fragmentation reduced application throughput by 50%, doubled the average number of disk commands queued for processing, and reduced idle CPU time.

Finding: Running Diskeeper with IntelliWrite® enabled over a one-week period, no file or free space fragmentation occurred.

Running Diskeeper with IntelliWrite enabled, all logical volumes performed at optimal optimization levels for the entire week. There was no new file or free space fragmentation on any of the system's disk volumes. In particular, IntelliWrite typically prevented 99% of file writes from fragmenting and instant file defragmentation removed any discovered fragments.

Benefit: All logical storage volumes performed at fully optimized levels for the entire testing period.

Running Diskeeper with IntelliWrite enabled, prevented any file space fragmentation, which in turn eliminated split I/Os from being issued. As a result, all physical I/Os were equal or greater than logical I/Os, which maximized application throughput. To further reduce the work imposed on storage devices, IntelliWrite's preallocation of file space enabled the Windows OS to bundle logical writes to further reduce the number of physical I/Os being performed by the storage system.

Benefit: By reducing the physical I/O workload on storage systems, Diskeeper maximizes the potential for datacenter energy savings.

From servers to laptops, power consumption has become increasingly important. In particular, machine density in datacenters has led to problems delivering sufficient power to devices and getting rid of the waste heat subsequently produced. Diskeeper is able to mitigate environmental and electrical cost concerns by measurably reducing the number of physical I/Os that a storage subsystem is required to perform to accomplish a fixed amount of work. Specifically, lowering the number of electro mechanical operations that a storage system performs during an active session directly lowers electrical power draw and the amount of waste heat that must be dissipated.

In addition, the complete elimination of scheduled defragmentation processes, which proved to be ineffective in the presence of severe fragmentation, compounded savings by eliminating the work needed to remediate the effects of fragmentation and extending periods of inactivity, during which storage subsystems can spin down drives and park disk heads.

Benefit: Under heavy processing loads Diskeeper easily continued to automatically prevent all fragmentation, which resulted in exceptional load balancing of I/O and CPU resources.

Diskeeper with InvisiTasking was able to sustain heavy I/O processing under IntelliWriteBench to automatically maintain exceptional I/O and CPU load balancing. On our server running Windows Server 2012, in particular, Diskeeper reduced the physical I/O transfers by 97%, increased the data contained in a single transfer operation by 42X, and doubled the percent of idle CPU time without any intervention by an IT administrator.

Executive Briefing:

Condusiv Diskeeper® 12 Intelligently Restructures Writes to Maximize Performance of Server and Desktop Systems

Jack Fegreus
October 8, 2012

Improving Storage Performance by Increasing I/O Throughput and Decreasing I/O Operations

The Critical Role of Disk Fragmentation

In this analysis, openBench Labs examines the ability of Condusiv Technologies' Diskeeper 12 to maximize I/O performance for server and desktop systems by restructuring the way data is written to disk. While the electro-magnetic technologies associated with computer processing and storage density continue improving at a 60% compound annual rate, the mechanics of disk storage, which are key to I/O throughput and access, have been advancing at a much slower pace. While storage density advances at a 60% compound annual rate, I/O throughput and access speeds have improved at a more modest 10% compound annual rate. As a result, disk system performance has become a critical factor in overall system behavior.

More importantly, the advances in processing power and storage capacity have spurred a revolution in the sophistication of business applications. From engineering design to financial trading, knowledge workers frequently use workstations and servers to run sophisticated calculations on large volumes of sparse matrix data to model and optimize complex processes, which have muscled into the ranks of mission-critical business processes at many enterprise sites.

For IT operations, this new class of mission-critical applications introduces a Pandora's Box of hardware and software issues. In particular, load balancing is essential for running these applications efficiently.

Load balancing techniques can set a trap for administrators attempting to resolve complex computational performance issues through the adoption of traditional load balancing schemes that narrowly focus on CPU and memory usage. Under I/O-intensive workloads,

Under Test: File System Fragmentation Prevention
Condusiv Diskeeper 12 Server & Professional

1) **Provide continuous I/O optimization:** By combining the prevention of file fragments with smart allocation and deallocation of logical blocks when creating or extending files, Diskeeper 12 with IntelliWrite® is able to provide a virtual storage space for every storage volume that does not require periodic defragmentation and does not degrade with time.

   Operations Benchmark: Running over a two-week period, IntelliWrite resolved 98% of file fragments (603,485 fragments prevented).

2) **Decrease physical disk accesses to a storage array:** By eliminating split I/Os to file fragments, Diskeeper 12 lowers physical I/O rates to disk, which improves overall throughput performance and extends the lifespan of storage resources.

   Iometer Benchmark: Running 128KB reads of a file fragmented without IntelliWrite-enabled generated at least twice as many physical I/Os per second with half the throughput rate in MB/sec.

3) **Relieve CPU bottlenecks:** By minimizing disk I/O operations and lowering the number of average commands in disk queues, Diskeeper 12 with IntelliWrite increases system idle time which returns available CPU cycles back to user processes.

   IntelliWriteBench Benchmark: Running IntelliWriteBench in a standard load configuration, the percent of system idle time was on average 50.8X higher with IntelliWrite enabled.

4) **Minimize system boot times:** Using HyperBoot Technology, Diskeeper 12 places files used at boot time for fast access during the startup process.

   BootRacer Benchmark: With HyperBoot enabled, the average time to boot into the Windows desktop was cut by 56%.

performance limitations in mechanically centric disk technologies often contribute to lower computational performance when traditional CPU- or memory-centric load balancing schemes are implemented. To remedy this problem, IT administrators need to focus on balancing storage resource loads to reduce response times of computationally intensive parallel jobs.

Nonetheless, understanding the importance of I/O load balancing in resolving computational problems is only the start for IT administrators on the road to taking control of the situation. An important first step is dispelling the myth that data fragmentation is concerned with the distribution of data on physical disk blocks. What makes data fragmentation important for I/O load balancing is the fact that it is probably IT's oldest storage virtualization construct.

"In this new IT environment, Diskeeper 12 Server and Professional have evolved to meet the next generation of streaming I/O requirements for multicore processors by preventing file fragmentation before it occurs."

Back in the Jurassic mists of IT computing, VAX behemoths running VMS dominated the IT landscape and the notion of a hardware abstraction layer (HAL) was generally acknowledged as essential for maintaining the efficiency of an operating system (OS) when dealing with the vagaries of proprietary hardware—particularly the geometry of disk drives. Long before virtualization would become an IT shibboleth, VMS decoupled physical storage space from a logical representation. By presenting a storage volume as a list of sequentially numbered logical blocks, a file could be represented as a simple string of logical block numbers (LBNs) containing the file data.

This scheme eliminated all of the complexities of track layout and bad block remapping on physical hardware. Nonetheless, it also introduced a new logical problem: an LBN file list would fragment when a file required more logical blocks than were sequentially available from its starting block. This introduced the concept of split I/O for files that were represented by disjointed strings of sequential LBNs. If an I/O request required data blocks from multiple LBN lists, the request had to be restructured into independent I/O requests for each list. To avoid this stumbling block in I/O efficiency, Diskeeper was originally introduced to restructure fragmented LBN lists as unified sequential lists in the virtual disk space created by VMS.

Fast forward to today's Windows Servers and PCs, which know no more about their underlying physical storage infrastructure than their minicomputer ancestors. As LBNs have given way to logical cluster numbers (LCNs) and desktop systems have acquired more computational firepower than previous generations of enterprise servers, the optimization of virtual I/O to maximize physical I/O throughput and IOPS access capabilities of current storage subsystems has dramatically increased in importance. In this new IT environment, Diskeeper 12 Server and Professional have evolved to meet the next generation of streaming I/O requirements for multi-core processors by preventing file fragmentation before it occurs.

In particular, when an application accesses a file on a system running any Windows OS, a query is sent to the Master File Table for the file's starting LCN. If the file is contiguous, only one logical I/O is required to read the entire file; however, if the file is not contiguous, the starting LCN of each contiguous segment of the file is needed and a logical I/O must be issued for each segment.

Consider a financial trading environment, where a high IOPS rate for transactions is the norm. Our IOPS benchmark tests demonstrate that fragmentation-related split I/Os increase the number of physical I/O operations that expensive fast-access storage subsystems have to perform in order to complete the work required by an application. The excess physical operations impact the array's I/O command queues, lower performance for all applications utilizing storage pools on the array, and add to the array's operating costs with respect to power and cooling.

Now consider an engineering scenario with large data files for applications from CAD to finite element modeling. Multiple fragments will require multiple reads and throughput will turn to sludge. Worse yet, the tests conducted by openBench Labs demonstrated that unless existing file fragmentation is eliminated entirely and free space is consolidated, new files will fragment rapidly and compound performance degradation on storage resources. As a result, IT will be pressured to expend real capital to upgrade storage I/O capabilities in order to resolve preventable virtual I/O overhead.

The key to the Diskeeper 12 value proposition has very little to do with periodic defragmentation and everything to do with the continuous prevention of fragmentation. Diskeeper 12 utilizes a number of light-weight processing technologies, such as InvisiTasking®, to avoid disturbing user application processes while continuously fostering high-speed I/O by invoking special features that can be assigned on a per volume basis.

> *"The key to the Diskeeper 12 value proposition has little to do with periodic defragmentation and everything to do with the continuous prevention of fragmentation."*

The most important Diskeeper 12 optimization feature is IntelliWrite, which proactively prevents file fragmentation by ensuring that files are created or extended as single contiguous collections of LCNs. A key part of the IntelliWrite process involves the smart preallocation of blocks for Windows files. Using a proprietary algorithm, IntelliWrite increases the normal space preallocation made by the Windows OS for the file and that in turn allows Windows to bundle multiple consecutive logical I/Os into a smaller number of what can be described as jumbo physical transfers.

Running a benchmark that generated logical 64KB I/O requests, we measured average physical I/O transfers to be 790KB per second. With Diskeeper Windows was able to bundle about 12 logical I/O requests into one physical I/O. Moreover, when a file was closed, Diskeeper released any unused logical blocks that were previously allocated. On the other hand, when we ran the benchmark in the same environment without Diskeeper, average physical I/O transfers plummeted to 62KB per second. Windows had to issue multiple physical I/Os for every logical I/O.

As a result, Diskeeper with Invisitasking continuously maintains optimal throughput and IOPS performance, which makes I/O performance more predictable and easier to manage. More importantly, IT no longer has to schedule periodic volume-level full defragmentation processes with the Windows built-in, which are CPU- and disk-intensive.

Continuous optimization with Diskeeper 12 includes the ability to transparently assign frequently used files to LCNs in the virtual storage space that are mapped to the fastest throughput region on the physical volume—whatever its underlying geometry may be. When an IT administrator enables I-FAAST® for a volume, Diskeeper 12 will test I/O performance during idle I/O periods to maintain a real performance profile of the volume. This performance profile—not a theoretical model of a simple disk—is used to guide the placement of frequently accessed files. What's more, Diskeeper 12 provides HyperBoot™ Technology to dramatically reduce the time needed to boot a system by locating boot files for accelerated access.

Proactive Fragmentation Avoidance Testing

Understanding the value proposition of Diskeeper 12, requires more than testing the ability to prevent the fragmentation of storage volumes. The full value of preventing data fragmentation is manifested in the way that it affects load balancing storage and CPU resources to maximize the performance of workstations and servers. As a result, we focused on four key issues for Diskeeper 12 in our test environments:

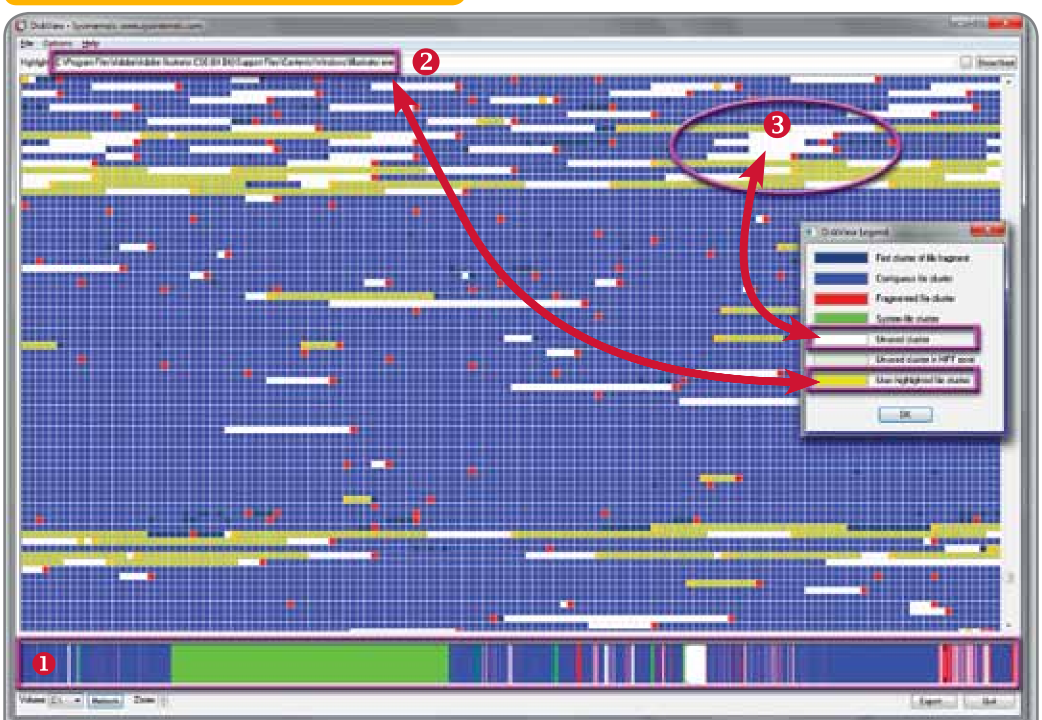
- **Provide a baseline for data fragmentation that occurs in Windows environments.**
- **Examine the ability of Diskeeper 12 to guide data placement to minimize split I/Os.**
- **Compare I/O performance with/without proactive defragmentation via IntelliWrite.**
- **Compare CPU defragmentation overhead with/without proactive defragmentation.**

To test both Diskeeper 12 Server and Professional, openBench Labs set up a Dell Precision T7600 workstation running Windows 7 and a Dell PowerEdge 1950 server running Windows Server 2012. These systems allowed us to configure nearly identical hardware environments, in which to compare and contrast the effectiveness of the Windows built-in defragmentation scheme on each system with Diskeeper 12's technology to prevent the fragmentation of storage volumes. In particular, the server and the workstation featured an Intel quad-core Xeon processor, 8GB RAM, a Dell PERC RAID controller, and four 6Gbps SATA drives configured as a RAID 5 array.

On both of the test systems, we used DiskView from Microsoft Sysinternals to analyze disk volume fragmentation. In addition, we used the Iometer benchmark to verify the effects of fragmentation on I/O performance at physical storage devices.

For any OS that does not have a way to prevent storage fragmentation before it occurs, the installation of new software will likely have an especially negative impact on the fragmentation of free space. To test the value proposition of a proactive fragmentation avoidance scheme in a workstation environment, we installed Windows 7, MS Office 2010, and Adobe Technical Communications Suite (TCS) 4, which supports the creation of sophisticated structured content using XML with FrameMaker, Illustrator, Captivate, and Acrobat Pro. Following the installation of these software packages and

Windows 7 Storage Fragmentation



After installing the workstation software, we used DiskView to examine the logical storage space of the workstation's system volume. DiskView provided a summary image ❶ of the storage space, as well as tools to zoom in on any area for a more detailed analysis and pick a logical cluster to discover its associated file ❷. In particular, the Illustrator application file—installed by Adobe TCS 4—was fragmented into 95 disjoint LCN collections. Worse yet, small free space fragments ❸ that were too small to fully encompass a work file for most applications littered the virtual storage space.

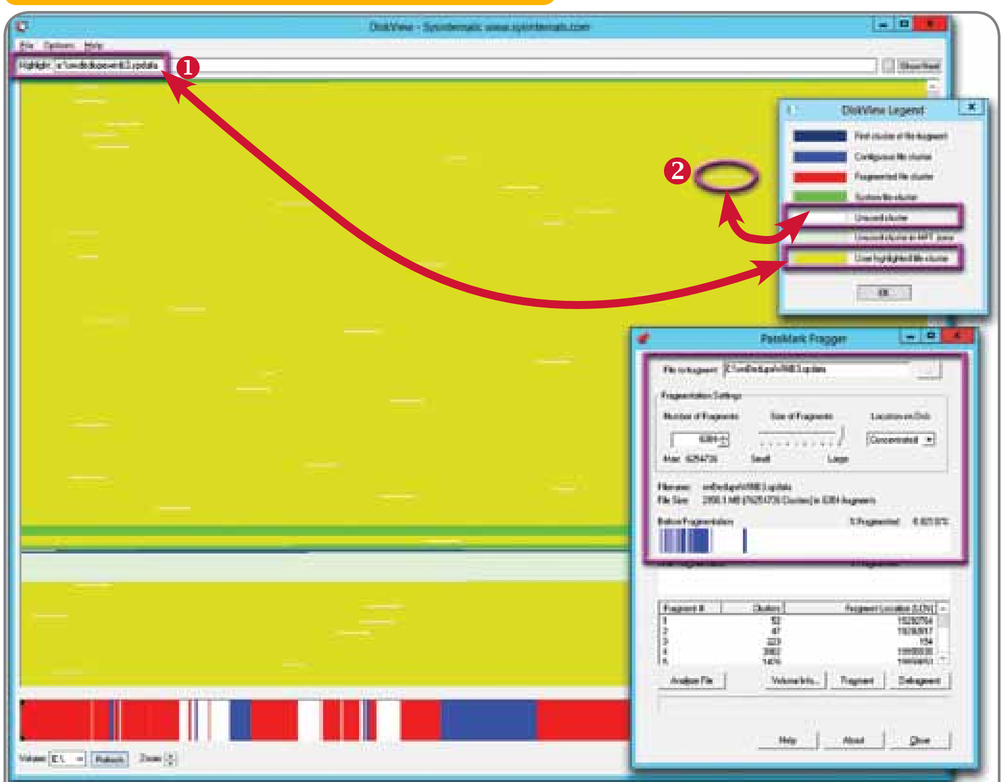
the updating of the OS and application software with feature and security patches, we examined the workstation's system volume with DiskView.

At issue was the placement of multiple large files. Even when the installation process can place every file in a set of contiguous logical blocks, it is highly likely that there will be small sets of contiguous empty logical blocks interspaced between application files. Those small interspaced free space fragments become serious stumbling blocks when application files are updated and extended. The interspaced free space is too small to support the new data, which means the logical layout of the application will naturally fragment. In particular, after running multiple instances of the standard Windows installation process and applying numerous software updates to both the OS and user applications, openBench Labs was left with a system volume on our workstation that was severely fragmented. Specifically we were confronted with:

- **A system volume with 2,026 fragmented files averaging five fragments per file.**
- **The most fragmented file had 906 fragments.**
- **Free space was fractured into 12,640 fragments with an average size of 296KB.**
- **The largest free space fragment was just 882MB.**

To assess fragmentation on a storage volume on a system running Windows Server 2012, we turned our attention to iSCSI SAN software, which many IT sites use to implement a highly functional SAN with direct attached storage (DAS) in a very cost-effective manner. Such a SAN is essential for supporting virtual machines (VMs) in a virtual infrastructure (VI) using VMware® or Microsoft Hyper-V.®

Windows Server 2012 Storage Fragmentation



On our StarWind server, we examined a DAS volume used for files representing logical iSCSI volumes that featured thin provisioning, snapshots, and global data deduplication. These files were characteristically large and exhibited rapid growth. As a result, each volume image ❶ file was highly fragmented with thousands of fragments, which were bounded by very small free space fragments ❷ that could not fully encompass any file used to support a volume feature.

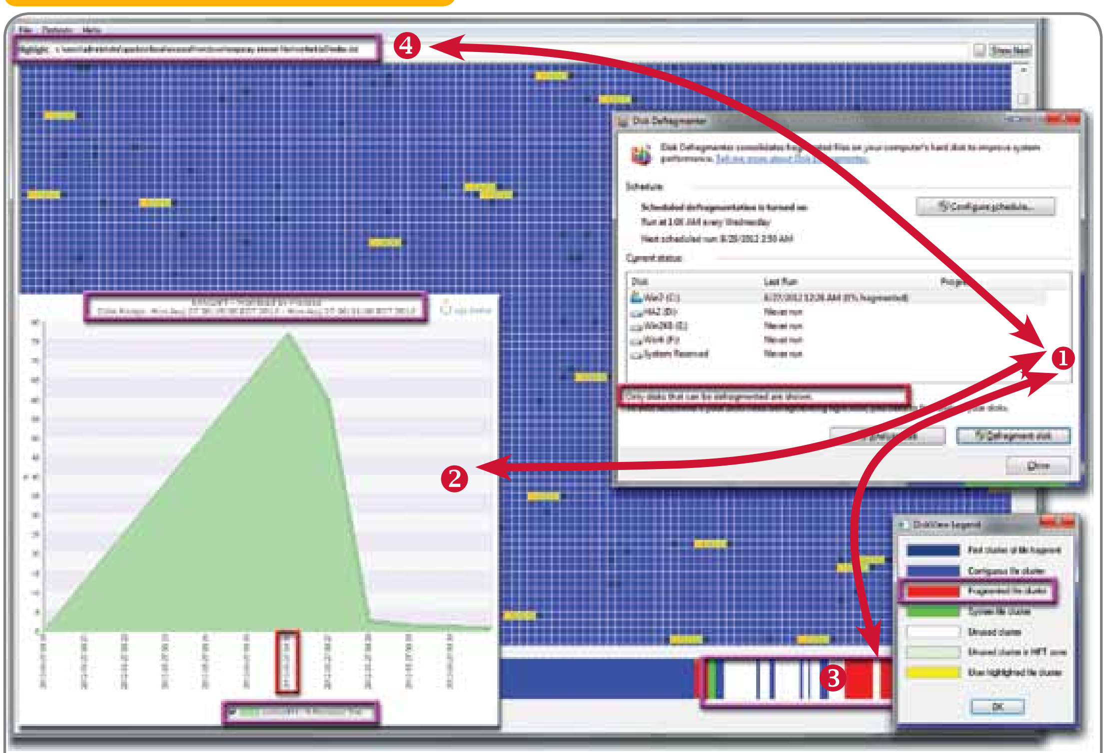
For our tests, we utilized StarWind iSCSI SAN software with local DAS volumes to provide advanced functionality, including thin provisioning, snapshots, global data deduplication, and 3-way high-availability failover, for logical volumes that we would expose as iSCSI targets. Given the StarWind enterprise feature set, vSphere and Hyper-V host servers are typically among the client systems supported by StarWind iSCSI SANs. As a result, IT administrators will typically populate storage volumes on servers hosting StarWind with very large NTFS files measured in hundreds of GBs if not TBs.

That container-file scenario creates a toxic brew that severely fragments the underlying virtual-block space of any volume supporting StarWind. As IT creates large image files, free space immediately starts to fragment. Then as users add data to thin-provisioned volumes and automated data-protection processes create periodic snapshots on the virtual volumes, the large container files on the StarWind server have to be extended into ever more fragmented free space.

> *"After running an intensive defragmentation process that ran over 10 minutes and consumed over 80% of a core processor, the built-in Windows 7 defragmentation utility reported that all file fragmentation had been eliminated; however, when we examined our system volume with Microsoft's Sysinternals DiskView utility, we discovered a storage space that was far from being completely defragmented."*

The process of creating and extending very large container files fragmented the logical storage space underpinning our server volume in a far more devastating manner than what we had witnessed on our workstation. We encountered files with an order of magnitude more fragments than any of the application files found on our workstation. In particular, we encountered server files that had fragments numbering in the thousands.

WINDOWS 7 DEFRAGMENTATION RESULTS

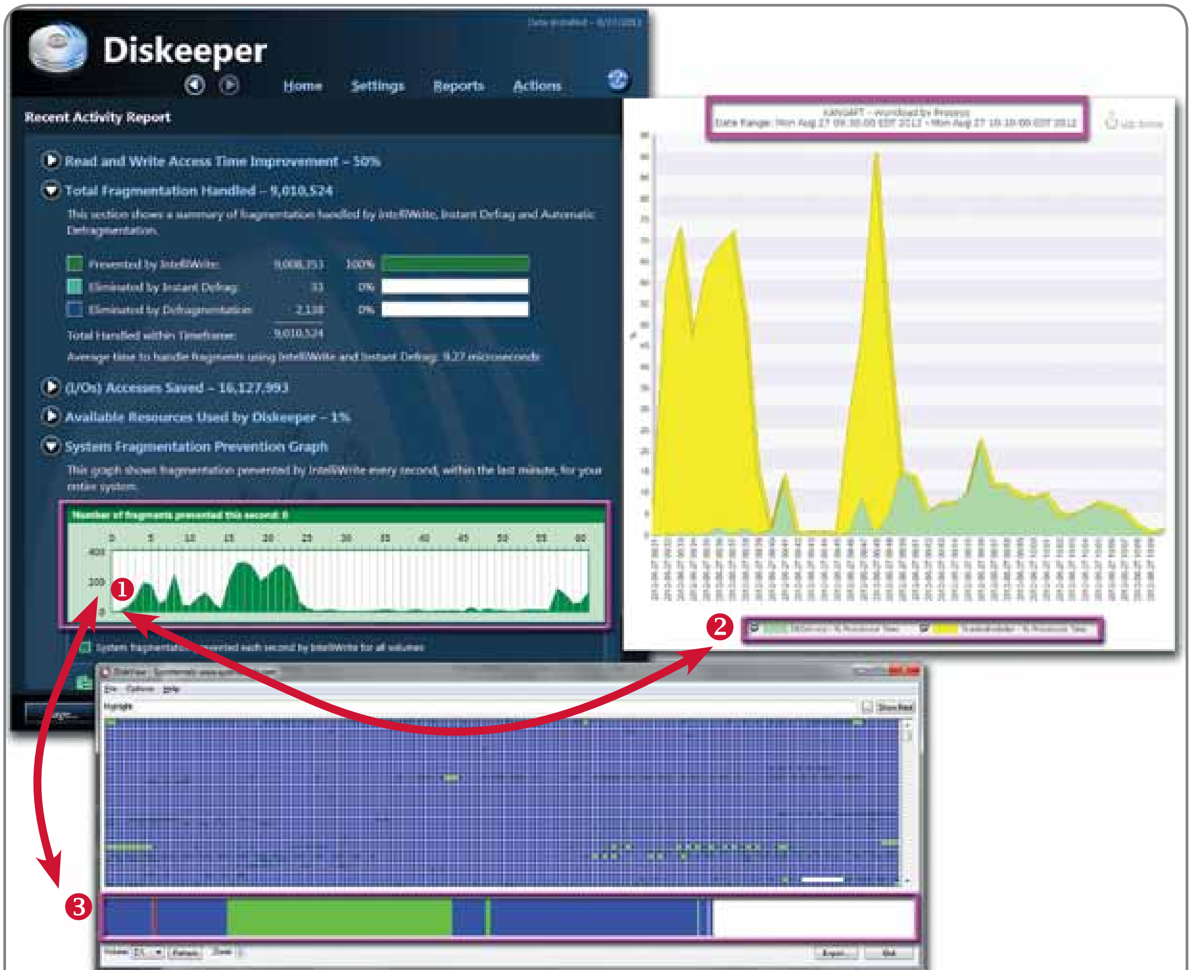


After the installation of our workstation software, we used the Windows 7 defragmentation utility and examined the logical storage space for the workstation's system volume. The defragmentation utility reported 0% fragmentation ❶ after running a CPU-intense process ❷ Nonetheless, an examination of the virtual storage space with DiskView indicated that significant file fragmentation ❸ remained. In particular, an index file ❹ still had over 30 fragments.

Before we attempted to remedy the problem of a fragmented storage space, we performed an image backup of the fragmented disk to have a standard scenario for comparing multiple platforms. Next we invoked the built-in utility bundled with the Windows OS to remediate the file and free space fragmentation that standard software installation processes had created on our system disk. This process was made all the more difficult by the Windows defragmentation utility, which provides minimal details regarding the fragmentation of files and free space and equally minimal feedback about the defragmentation process.

After running an intensive defragmentation process that ran over 10 minutes and consumed over 80% of a core processor, the built-in Windows 7 defragmentation utility reported that all file fragmentation had been eliminated; however, when we examined our system volume with Microsoft's Sysinternals DiskView utility, we discovered a storage space that was far from being completely defragmented. On a closer look, we found multiple files that had 30 or more disjointed fragments distributed across the volume's logical storage space. What's more, we discovered that the volume's free space had not been consolidated, but remained highly fragmented and interspersed between clusters of file fragments and files that had been defragmented by the built-in Windows utility.

Fragmentation Avoidance Results



During the installation of Adobe Technical Communication Suite, we observed peak IntelliWrite fragmentation prevention ❶ to exceed 200 fragments per second. At the same time, overhead for the Diskeeper 12 service ❷ typically ranged from 5-to-10% with occasional spikes up to 20%. Following the installation of our software there was no need to follow up by running a full defragmentation of the target disk, which was in excellent condition ❸ with respect to free space and future I/O operations.

By not addressing future data file fragmentation at the root cause, which is dispersed and fragmented free space, the periodic defragmentation of files using the built-in Windows utility would at best be a placebo. Given the state of our system volume's virtual storage space, the creation of any new file—especially a large one—inevitably resulted in the addition of another highly fragmented file, which would act as a catalyst for even more fragmentation.

In particular, when we ran the Iometer Dynamo utility to create a 1GB test file, with which we could measure throughput for this volume, we consistently ended up with a test file having about 200 fragments. While periodic defragmentation with free space consolidation may slow the degradation of I/O performance, periodic defragmentation cannot stop performance degradation—let alone reverse the process.

Defragmentation Problems for Today's IT

Next, we repeated the installation of Windows 7 and the requisite application software; however, this time we stopped the process immediately after installing the OS. At that point, we installed Diskeeper 12 Professional and enabled IntelliWrite to proactively avoid fragmentation. We then monitored the IntelliWrite process with respect to CPU overhead and I/O throughput over the rest of the installation process.

When we completed the required updates to Windows 7, along with all of the applications software, including Adobe Technical Communications Suite, and MS Office, an examination of the storage space for the system drive revealed it to be in a near-perfect state. This was in sharp contrast to the state of our system disk after running the built-in defragmentation utility. Not only were file fragments nearly impossible to find, free space too showed minimal fragmentation.

IntelliWrite Reporting

Using the Diskeeper 12 reporting facility, we were able to track fragmentation prevention and remediation over time. Over an initial two-week period, IntelliWrite prevented over 603,000 file fragments. No defragmentation periods were scheduled following the initial defragmentation of 12,000 file fragments.

More importantly, whether we were installing daily software updates or new software packages, IntelliWrite instantly acted to ensure that all files were contiguous. What's more, the IntelliWrite process consistently required very minimal CPU processing, even when it was necessary to prevent the formation of several hundred fragments per second during a complex installation process for a large application package.

As a result, Diskeeper easily maintained the logical storage space for all of the drives on a server or a workstation with IntelliWrite enabled. Over a period of two weeks that followed an initial manual defragmentation, nearly all fragments were handled via IntelliWrite. Over 603,000 file fragments were prevented from forming. Using InvisiTasking, Diskeeper only needed to automatically defragment a meager 787 fragments to keep the volume in an optimal state.

If we had been following a traditional periodic defragmentation strategy with the built-in Windows utility, we would have needed to defragment upwards of 600,000 fragments manually. To contend with that volume of file fragmentation and avoid

significant performance degradation, nightly defragmentation sessions would have been required. More importantly, the effectiveness of Diskeeper with IntelliWrite has profound implications for efficient backup and the realization of green energy savings in a VI.

For more efficient backups VMware hosts employ a technique dubbed Changed Block Tracking (CBT) for each VM. CBT limits the data transferred in an incremental backup to changed blocks rather than changed files. By moving files to consolidate free space, defragmentation changes a plethora of blocks simply by moving them, which negates most of the efficiencies gained with CBT.

Regularly scheduled defragmentation processes also negatively impact best practices for green energy savings. A common strategy for lowering power consumption of storage arrays spins down idle drives in a scheme dubbed Massive Array of Idle Disks (MAID). In a traditional infrastructure, regularly scheduled defragmentation does not disrupt MAID savings as the number of physical servers sharing storage arrays—whether through dual porting, iSCSI, or a Fibre Channel SAN—is relatively small. Best practices for load balancing in a VI, however, call for maximizing shared storage for easy movement of VMs among hosts.

Moving VMs requires VI hosts to share access to storage arrays that contain VMs. As a result, multiple VMs running local defragmentation processes will keep a common set of storage arrays busy over extended periods and that will significantly diminish any energy savings from a MAID scheme.

Nonetheless, interruption of idle time by defragmentation processes is only a small part of the negative impact that file fragmentation can have on energy consumption. While current best practices for green energy savings concentrate on idle activity periods, most industry research is now focused on minimizing power draw during active I/O periods.

An important catalyst for this new line of focus is the need to extend battery life for laptop computers by cutting the power consumption of its hard drive by at least 5X. Another catalyst is the problem of I/O storms that occur when multiple VMs increase their I/O loads in concert and dramatically lower processing across the entire VI. As a result, any process that increases I/O strictly as an overhead factor, has the potential to significantly lower I/O performance across an entire VI.

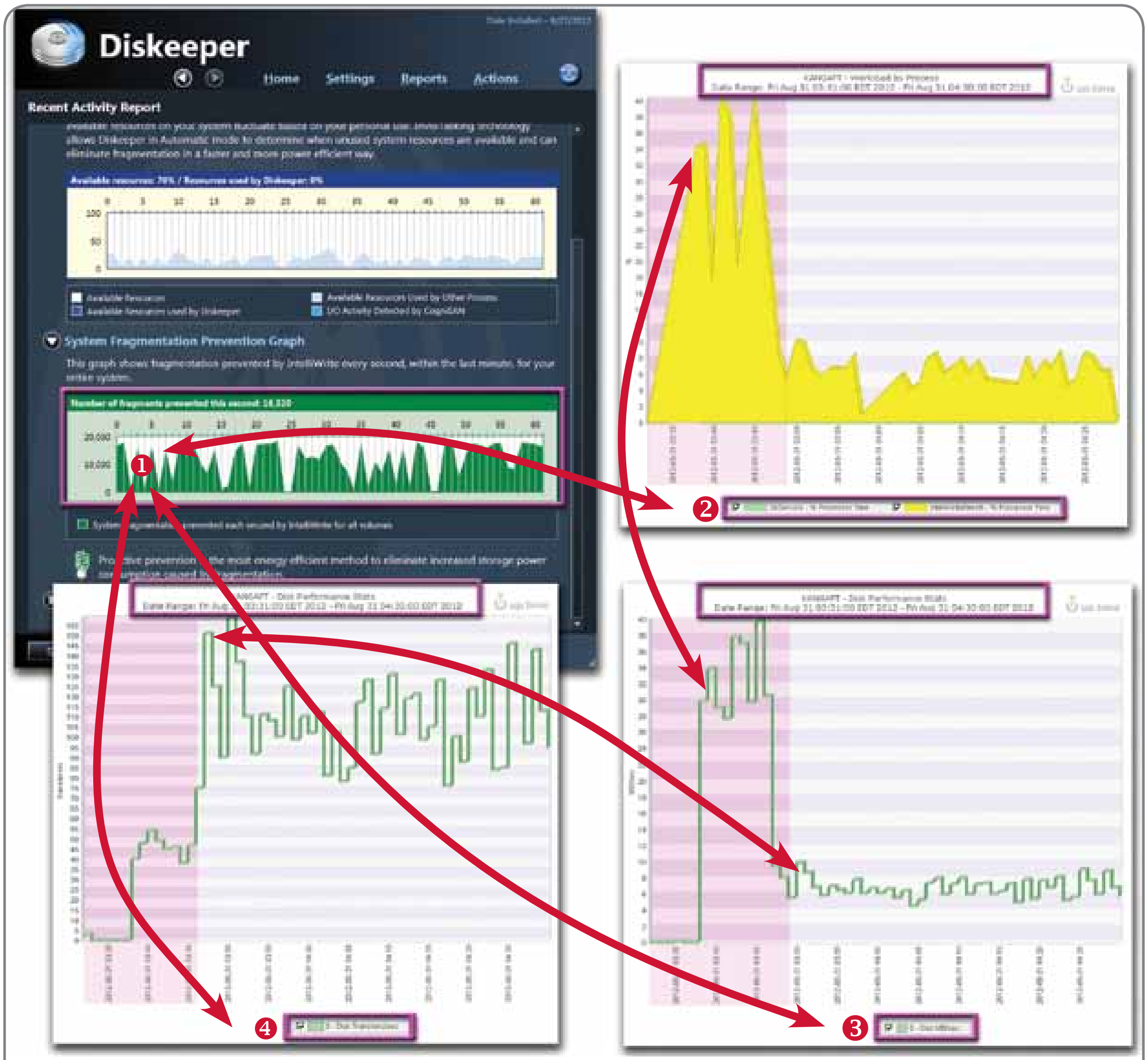
The Real Value of Fragmentation Prevention

To gain deeper insight into the ability of IntelliWrite to change I/O dynamics we ran IntelliWriteBench on both a desktop workstation running Windows 7 and a server running Windows Server 2012. Using a fixed workload based on a specified test profile, this benchmark simulates user activity by creating and then adding data to a number of test files. In this scenario, creating test files and then adding new data increases their size, which stresses the underlying storage space with a high probability for creating fragmented files.

In particular, IntelliWriteBench runs three iterations with Diskeeper 12 enabled and then three iterations with Diskeeper 12 disabled. On the second set of three iterations, IntelliWriteBench simulates performance in a traditional storage environment where optimization is focused on periodic defragmentation to remediate file and free space fragmentation. During each of the IntelliWriteBench iterations, IntelliWriteBench uses Perfmon and Microsoft Sysinternals Contig to collect and analyze processing data, including the number of file fragments generated.

We also monitored the processes running on our test system along with the I/O on our local system disk using a LAN-attached server running the up.time software monitoring package for a second perspective. Over the testing period, we observed IntelliWrite typically handling between 15,000 and 25,000 file fragments per second, which was 10X greater than the peak rates that we measured while we installed our application software.

IntelliWriteBench Fragmentation Prevention



Running IntelliWriteBench, IntelliWrite was presented with processing about 15,000 file fragments ❶ per second, which is an order of magnitude greater than we had measured during our software installation tests. While running the benchmark, we measured CPU utilization ❷ for IntelliWriteBench—the simulated user process—and the Diskeeper 12 service. We also measured I/O throughput ❸ and disk IOPS ❹ for the disk used in the benchmark. With IntelliWrite enabled, I/O throughput was 5X greater, CPU process utilization for IntelliWriteBench was 4X greater, and wall clock processing time was 4X quicker. When IntelliWrite was disabled, physical I/O operations rose by a factor of 2.5X, while I/O throughput dropped to less than 20% of the level sustained with IntelliWrite enabled.

From all of the data associated with the IntelliWriteBench process, a number of critical factors emerged involving CPU resources and the number of physical I/O operations. What's more, these factors dramatically change in an environment where file and free space fragmentation is prevented from occurring.

Specifically, when Diskeeper was disabled, we measured a dramatic 10X rise in the rate of virtual split I/Os on our fragmented workstation disk along with 2.5X rise in the rate that physical disk I/Os were issued. More importantly on an empty server volume, we measured a 66X rise in the rate of virtual split I/Os and a 31X rise in the rate of physical disk I/Os being issued. Without IntelliWrite, a greater workload was imposed on physical disks without regard to whether or not the disk was fragmented. In turn, those excess physical I/Os burdened all applications using the array. More importantly, without

IntelliWrite adjusting the preallocation of file space, the average size of a data transfer shrunk by a factor of 12.5X on our workstation and 42.7X on our server. As a result, the data throughput rate with Diskeeper on our workstation was 5X greater and 1.3X greater on our server with a newly formatted disk. Clearly, Diskeeper was able to improve I/O performance in any environment.

IntelliWriteBench Results
Diskeeper 12 Server and Professional

Performance Data Metric	Iteration	Diskeeper 12 Server Enabled	Windows Built-in Enabled	Diskeeper 12 Performance Improvement	Diskeeper 12 Professional Enabled	Windows Built-in Enabled	Diskeeper 12 Performance Improvement
Split I/Os per Second	Test 1	19.8	1,147.4	98% fewer I/Os	20.3	113.7	81% fewer I/Os
	Test 2	17.6	1,086.4	98% fewer I/Os	8.0	103.3	92% fewer I/Os
	Test 3	17.1	1,114.4	98% fewer I/Os	7.1	107.8	93% fewer I/Os
	Average	18.0	1,182.7	98% fewer I/Os	11.8	108.3	89% fewer I/Os
Physical Disk Transfers per Second	Test 1	35.6	1,153.3	97% fewer I/Os	49.6	116.2	57% fewer I/Os
	Test 2	38.3	1,033.3	96% fewer I/Os	44.5	105.7	58% fewer I/Os
	Test 3	39.2	1,321.7	97% fewer I/Os	45.0	110.4	59% fewer I/Os
	Average	37.9	1,189.4	97% fewer I/Os	46.4	110.8	58% fewer I/Os
Data KB per Transfer	Test 1	3,049KB	64KB	47.6X more data	690KB	62KB	11.1X more data
	Test 2	2,627KB	64KB	41.0X more data	753KB	63KB	12.0X more data
	Test 3	2,696KB	64KB	42.1X more data	846KB	61KB	13.9X more data
	Average	2,731KB	64KB	42.7X more data	790KB	62KB	12.7X more data
Disk MB per Second	Test 1	100.0MB/s	75.1MB/s	1.3X higher MB/s	33.4MB/s	7.5MB/s	4.5X higher MB/s
	Test 2	97.4MB/s	71.2MB/s	1.4X higher MB/s	38.2MB/s	6.7MB/s	5.7X higher MB/s
	Test 3	97.6MB/s	85.9MB/s	1.1X higher MB/s	38.2MB/s	7.2MB/s	5.3X higher MB/s
	Average	98.3MB/s	77.4MB/s	1.3X higher MB/s	36.6MB/s	7.1MB/s	5.2X higher MB/s
Average Disk Queue Length	Test 1	15.8	81.3	81% fewer I/Os	7.8	25.0	69% fewer I/Os
	Test 2	15.2	98.0	78% fewer I/Os	5.7	26.4	78% fewer I/Os
	Test 3	14.5	84.4	85% fewer I/Os	6.2	24.1	74% fewer I/Os
	Average	15.1	87.9	83% fewer I/Os	6.6	25.2	74% fewer I/Os
Percent Idle Time	Test 1	4.14%	1.73%	2.4X greater	2.16%	0.13%	16.6X greater
	Test 2	2.54%	1.73%	1.5X greater	2.69%	0.01%	269X greater
	Test 3	2.77%	1.34%	2.1X greater	2.77%	0.01%	277X greater
	Average	3.15%	1.60%	2.0X greater	2.54%	0.05%	50.8X greater

An increase in physical I/Os also has a deleterious effect on consumption of electric power by the storage subsystem. Increased physical I/O operations correspond to more electro-mechanical work, such as disk arm movements needed to reposition magnetic read/write heads. As a result, the drive draws more power and generates more heat.

What's more, electric power consumption effects from an increase in disk workload scale with the size of the environment. For a laptop, the workload issue will likely mean a lower operating period for a given battery charge. For multiple servers in a large datacenter with a highly integrated storage infrastructure, however, the effects of increased physical I/Os on power and cooling can be quite costly.

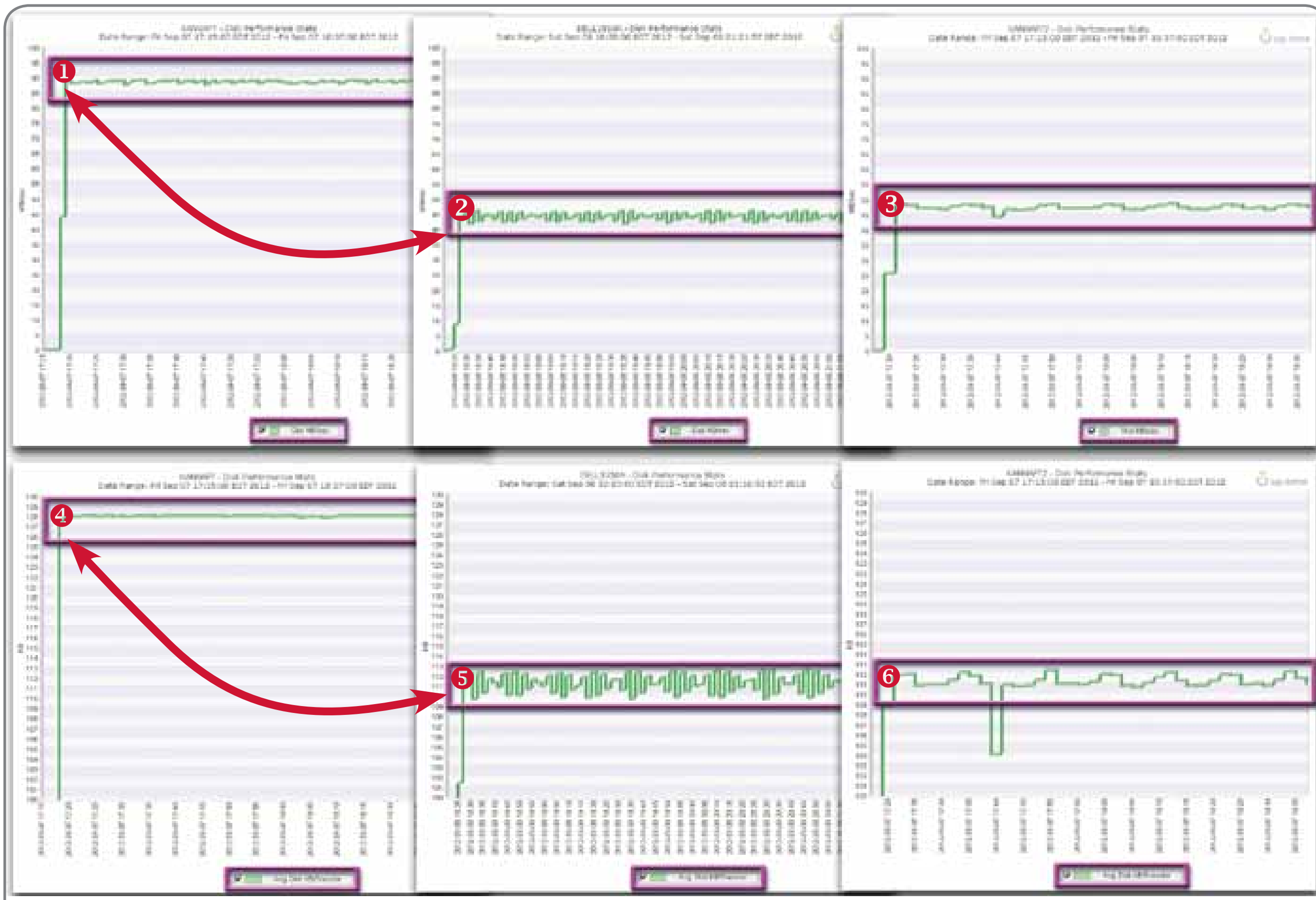
The IntelliWriteBench results also cast a critical light on load balancing. Both the workstation and server were significantly impacted by the I/O processing time required to run the benchmark without IntelliWrite. In particular, the percent of CPU idle time dropped to near zero leaving no CPU cycles available for applications. Under the workload created by IntelliWriteBench, turning IntelliWrite off transformed an I/O bottleneck into a CPU bottleneck and left our test systems CPU- and I/O-bound.

While IntelliWriteBench revealed a number of key performance issues, it did so by flooding a system with an extraordinary workload. We used Iometer to test fragmentation effects with a more typical workload on both the server and the workstation. For this test, we mounted images of the disk that we had fragmented during the installation of Windows 7 and user applications.

"Under the workload created by IntelliWriteBench, turning IntelliWrite off transformed an I/O bottleneck into a CPU bottleneck and left our test systems CPU- and I/O-bound."

To begin Iometer testing, we attempted to create a test file that was not fragmented. On both the server and the workstation, we used Iometer's Dynamo utility to create a 1GB file. Without IntelliWrite enabled, however, dynamo consistently created files that had about 200 fragments.

Iometer Throughput & Physical I/O



We configured Iometer to read a 1GB test file sequentially using 128KB blocks. On a workstation disk with IntelliWrite enabled, we were able to create a contiguous file and read data at about 90MB ❶ per second. Without IntelliWrite support, we had to use a test file that had 197 fragments. Using this file, throughput on both the server ❷ and the workstation ❸ dropped to about 45MB per second. Virtual storage space fragmentation clearly lowered real throughput; however, a different metric was needed to show it also caused multiple physical I/O requests to be generated to fulfill the logical 128KB requests from Iometer. To show that multiple were required to fulfill the 128KB requests, we examined the average size of a data transfers per second. With a contiguous file, data transfers were consistently 128KB ❹ per second. With a fragmented file, both the server ❺ and the workstation ❻ fluctuated between 110KB and 112KB per second.

Next, we compared the performance of Iometer as it read the test file sequentially using 128KB blocks, which are typically used by backup and other applications that need to read large amounts of data. Running Iometer with a contiguous file, we were able to sustain reads at 90MB per second. On the other hand, throughput for Iometer reading a fragmented file dropped dramatically to about 45MB per second on both our server and workstation.

We also examined the average size of data transfers per second. Using our contiguous file, the size of physical I/Os averaged 128KB per second, which was precisely the size of the logical I/O requests being issued by Iometer. Using the fragmented file, the average data transfer size fluctuated between 110KB and 112KB per second. That meant an average of at least two physical I/Os of different sizes were being issued to fulfill each 128KB request issued by Iometer.

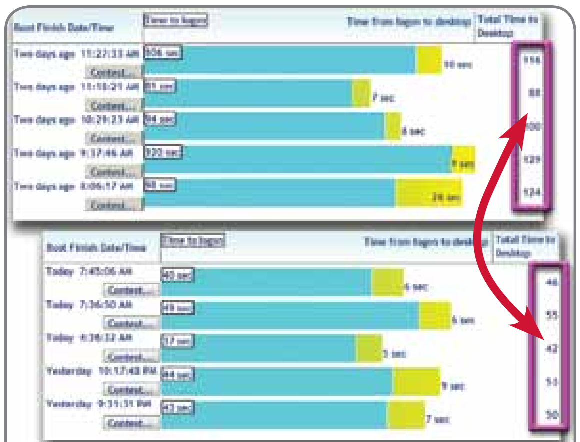
More importantly, the dramatic difference in throughput indicates that the multiple I/O requests regularly required disk track changes, which are the most expensive movements. The time required to complete a disk seek that moves from one track to another significantly dwarfs the time needed to transfer data from the disk. What's more, track-to-track seeks dramatically increase the electro-mechanical work being done by the drive.

Smart Data Placement

In addition to preventing file fragments from being written to disk, Diskeeper 12 Professional provides IT with a HyperBoot feature that creates a system-specific boot plan that utilizes sophisticated caching and knowledge of the performance of the Windows system disk. The HyperBoot plan is designed to accelerate loading the files Windows needs to boot in a way that minimizing costly disk seeks when reading all of the current boot files.

As with file fragmentation, boot time is extended by changes and additions to applications and user profiles that modify the data that must be loaded when a user boots into the Windows desktop. To resolve this problem, Diskeeper 12 Professional tracks all ongoing boot times and uses this data to automatically update the system's boot plan. In this way Diskeeper 12 is able to continuously optimize a minimal boot-up time on any desktop PC running Windows 7, 8, or XP.

HyperBoot Time To Desktop



Using BootRacer utility for Windows, we timed a series of boot-up tests of our workstation with and without the HyperBoot facility in Diskeeper 12 Professional enabled. Using HyperBoot, the average time needed to boot our workstation to the Windows desktop dropped by 56%.

For IT the problem with extended boot time is two-fold. First it reinforces a perception among end users that there is a performance problem that is not being addressed by IT, which negatively impacts IT's corporate reputation. Second there is the classic problem of I/O storms when all of the users at a site tend to boot their desktop systems in a narrow time frame or IT attempts to automatically update software on desktops across a site. PCs running Windows 7 frequently require a reboot following the installation of new software.

Throughout a series of boot-up tests, which we monitored with the independent BootRacer utility, HyperBoot proved to be a very simple way to dramatically improve the PC startup experience. Specifically in our tests, we were able to reduce the time to start Windows and login to the desktop environment by 56% without making any manual configuration changes to Diskeeper 12 or the Windows 7 OS.

Customer Value

For IT operations, growing numbers of mission-critical business processes introduce a Pandora's Box of hardware and software issues. Load balancing IT resources presents IT administrators with complex computational performance issues. By adopting a traditional load balancing scheme that narrowly focuses on CPU and memory usage, IT administrators risk overlooking root causes that are I/O related. Under I/O-intensive workloads, mechanically-centric disk technology needs to be carefully optimized to avoid becoming a sink for what should be idle CPU cycles for use by applications.

openBench Labs Disk Optimization Testing Summary

Feature	Key Test Analysis	Condusiv Diskeeper 12	Windows Built-in Defragmentation
Automated Continuous I/O Optimization	New files and extensions of existing file are written using contiguous LCNs.	Diskeeper implements two tiers of automated fragmentation protection: prevention and instant defragmentation.	Periodic removal of file fragments risks the build up of free-space fragments and trigger geometric growth of file fragmentation.
Minimize Physical I/O Operations	Excess physical operations lower I/O throughput (MB/sec) for the application and adds overhead to other applications dependent on the storage array.	In addition to preventing file fragmentation, Diskeeper intelligently increased the space preallocated for writing a file which allowed the Windows OS to bundle logical I/Os into jumbo data transfers.	Between defragmentation processes, new files and free space continued to fragment and multiple physical I/O requests were needed to satisfy large-block logical I/Os, which were frequently split.
Relieve CPU Bottlenecks	Excess physical I/Os extend I/O command queues and consume available CPU cycles that would otherwise be available for applications.	Running I/O intensive tasks with Diskeeper minimized the length of disk command queues and consistently left measurable CPU idle time.	As multiple physical I/Os extended the time required to process logical I/Os, the I/O throughput rate dropped precipitously and we also measured a similar drop in the availability of idle CPU processing cycles.
Minimize Desktop System Boot Time	For automated IT processing of software installations and updates, faster reboots help to minimize maintenance windows.	The Diskeeper HyperBoot utility cut average boot times on desktop systems by 50-to-60% and continued to monitor and modify the files needed for booting as the system configuration changed over time.	Workstation boot time expanded over time.
Maximize Green Energy Savings	For active I/O processes, physical I/O is reduced which reduces power consumption by electro-mechanical components and heat dissipation. For idle I/O processes, periodic defragmentation processes impinge on MAID schemes which attempt to lower power consumption by spinning down disks during inactive periods.	Diskeeper was able to fully eliminate fragmentation without running costly defragmentation processes. Furthermore, through aggressive preallocation of space, the Windows OS was able to bundle large-block I/O requests into multi-megabyte jumbo requests to further cut physical I/O.	Periodic runs of the Windows defragmentation utility failed to eliminate all file fragmentation and did little to consolidate free space. Between defragmentation processes, file fragmentation grew progressively worse, which increased split logical I/Os by an order of magnitude, generated 2.5 physical I/Os for every logical I/O, cut throughput by 50%, and consumed all available idle CPU cycles to process excess I/O requests.

To remedy this problem, IT administrators need to focus on balancing storage resource loads to reduce response times of computationally intensive parallel jobs. In particular, disk fragmentation is the major cause of split I/O which translates into significant increases in physical I/O operations to read or write a fixed amount of data. As a result, disk queues fill up with excess commands that must be processed, more processing time is expended handling I/O, and less CPU resources are available to run user applications.

In particular, the number of physical I/Os per second on a volume optimized with Diskeeper were 58% less when handling the same data, which had remained fragmented after the built-in Windows 7 defragmentation utility was run. Moreover, user data throughput rates were more than twice as fast when reading from the optimized volume compared to the defragmented volume.

Diskeeper 12 resolves these problems not by defragmenting files, but by continuously preventing fragmentation before it occurs. As a result, Diskeeper is able to maintain a continuously optimized file structure without running full volume defragmentation processes. This is critically important for VMs in a VMware VI. VMware hosts optimize backup processes using a CBT mechanism. Volume-level defragmentation moves files in a way that often renders a large number of moved blocks as having been changed for the purpose of backup.

In addition, Diskeeper aggressively preallocates larger amounts of space for writing files based on file type. Diskeeper's preallocation process enables the Windows OS to bundle logical I/Os into even larger jumbo physical I/O transfers that further reduce physical I/O operations. Furthermore, Diskeeper truncates any over-allocation on closing the file so there is no wasted space. As a result, users have a more responsive desktop computing environment, and IT can delay many costly CPU, memory, and storage upgrades.

Diskeeper also utilizes its knowledge of the structure of the system disk on desktop and laptops to constantly maintain accelerated boot times. In our tests, boot time on multiple desktop systems running Windows 7 was typically cut in half.

For IT, these changes directly impact operating costs in a number of intriguing ways. By reducing physical—not just logical—I/O requests, there is an immediate impact on the work being performed by storage subsystems, which directly translates into green energy savings. More importantly, in a shared storage environment, especially in a VI environment, a reduction in physical I/O requests directly lowers the overhead on processes across multiple VMs using storage on common arrays.

Westborough, Mass.-based openBench Labs was founded in 2005 by Dr. Jack Fegreus. openBench Labs is a trusted IT industry source, providing hands-on evaluation and certification of Information Technology products and services. openBench Labs enjoys a unique position in the information technology sector. As the premier independent test lab and provider of third-party validation services, OBL has worked with virtually every major vendor and evaluated the most important products and technologies to appear over the past decade.