

*How fast are your files?*TM

A Case for Monitoring File I/O Performance

Tom West

TomWest@hyperIO.com
hyperI/O LLC.

July 2001 (revised Oct 2008), WP-1177-15

Abstract

Amidst the unabated advance of faster computer hardware, the explosive growth of storage, and the advent of virtualized storage and systems becoming more commonplace within the commercial setting, the performance of storage I/O has undergone even greater scrutiny. Yet surprisingly, the basic ability to readily assess (specifically, measure and monitor) the performance of storage I/O operations *at the file level* (particularly upon an individual application basis) has been long neglected. This absence is particularly astonishing given that much (if not most) of the actual work being performed by (faster) computer systems requires significant file I/O operation activity.

We contend that file I/O performance monitoring is a matter of practical necessity and growing importance. We also propose and briefly describe hIOmonTM. Featuring a flexible, adaptable and scalable architecture, this novel, generally available implementation of a file I/O performance monitoring facility affords a multitude of benefits to a wide audience (including researchers, developers, consultants, vendors and computer end-users alike). These benefits include salient improvements in the ability to obtain empirical I/O operation performance data for better design decisions; better diagnose and understand storage access performance problems; evaluate proposed improvements to the performance of computer systems; verify and ensure that required levels of file I/O performance (QoS) are being met; help reduce storage management costs, and prudently approach and adopt emerging storage technologies.

1. Introduction

How fast are your files? A simple and seemingly straightforward question, but one that is not readily nor often directly answered, especially in a timely, precise, and meaningful manner. Moreover, this is in distinct contrast to many other performance aspects of computer systems (as examples: processor speed, disk rotation speed, maximum bus and interface transfer rates, average seek time of a particular disk drive, etc.). So whereas one might proudly proclaim the faster ‘revolutions per minute (RPM)’ speed of their latest disk drive purchase, the corollary question as to exactly how much faster their particular files are actually accessed (as a result of their new disk drive purchase) will, unfortunately, most probably go unanswered.

Furthermore, a faster disk drive/subsystem is usually not acquired simply to have a speedier disk drive/subsystem (except perhaps by some “power-user” computer aficionados). Rather, we expect and mostly assume that the faster disk drive/subsystem will improve the data transfer performance of the files that reside upon the disk drive/subsystem. The end goal is the improvement of the file I/O performance. This improvement is sought-after due to the essential role that files play within computer systems

The prior discussion suggests two points. Firstly, the question “*How fast are your files?*” is in general a problematic one. Secondly, the ability to satisfactorily answer this question is important since it deals with a significant component of the computer system (namely, files); accordingly, considerable benefits can ensue from such a competency.

This paper makes a case for monitoring the performance of storage I/O operations at the file level. We begin by demonstrating the need for such a capability, and then describe some of the substantive benefits that such a capability could afford (and thus its growing importance). The next section of the paper outlines basic requirements of a file I/O performance monitoring facility. Lastly, we briefly discuss several past and present efforts to implement various file I/O performance monitoring capabilities. hIOMon, a novel implementation of a file I/O performance monitor, is also briefly introduced.

Overall, the key contributions of this paper include the revelation that files, although a crucial component of computer systems, have nonetheless essentially gone unmonitored to date, and additionally, the proposal of an innovative file I/O performance monitoring facility to address this significant shortcoming.

2. Why Monitor File I/O Performance?

In this section of the paper we establish the rationale for monitoring the performance of storage I/O operations at the “file” level and describe some of the derivative benefits.

2.1 Background

The drive to develop, provide and utilize faster computer hardware continues unabated. Ongoing improvements in hardware to increase the speed and overall performance of computer systems are fairly widespread and well known. One of the more prominent and advertised areas of improvement is microprocessor clock speed, with processors available having clock speeds of much greater than one gigahertz (1 GHz). The number and type of other improvements abound; a short litany might include larger addressability, faster/wider system bus, increased processor caching capabilities, larger/faster physical memory, the PCI-X [PCI-X] and PCI-E protocols, the InfiniBand™ architecture [INFTRADE], transports [GILDER], Ethernet [10GBE], SCSI [SCSI-7], and Fibre Channel [FCIA].

It is apparent that making computer hardware faster (i.e., increasing the speed of the computer hardware, typically by a significant amount – often doubling) is a primary goal behind many of these improvements. Current performance capabilities already operate within the “giga” range, with the expectation that future computer hardware will continue to exhibit substantial (if not astonishing) performance gains¹.

On another front, the drive to increase storage capacity (both in terms of the individual storage device and the acquisition of storage at large) also continues unabated. The explosive growth of storage can be viewed from two perspectives: from the viewpoint of the storage device itself (i.e., the increased capacity of the individual storage devices) and from the viewpoint of the growing amount of storage deployed globally.

As regards the storage devices themselves, storage capacities in some cases have doubled each year (which actually exceeds Moore’s Law) [GRAY1, MASHEY]. However, while magnetic disk capacity has improved dramatically, the associated rate of speed improvements (e.g., access rate and transfer rate) has not kept pace [GRAY1, GRAY2]. Consequently, the overall performance penalty associated with disk accesses has potentially increased. In addition, as disk and tape capacity approach infinity, the cost/GB metric essentially goes to zero with the cost/access metric becoming a more dominant performance metric [GRAY3].

From the macro level viewpoint, the dramatic and continuing increase in storage device capacity has been complemented with an explosion in storage procurement and use. While the growth rates vary considerably, digital data growth in some cases appears to be growing in excess of 100% annually [MOOREF2]. Other studies further confirm the tremendous growth rate of data storage [BRECHR, WHITING, and LYMAN] and the associated expenditures [BRECHR, DERRS].

¹ According to Moore, “By the year 2012, Intel should have the ability to integrate 1 billion transistors onto a production die that will be operating at 10GHz. This could result in a performance of 100,000 MIPS, ...” [MOOREG].

Certainly the enormous and ongoing growth of storage (both at the individual device level and the aggregate deployment level) incurs grave ramifications as to the attendant storage management (particularly when the cost of managing storage is 3 to 10 times greater than the cost of the hardware, based upon current industry findings) and as to the impact upon overall storage I/O performance. Storage connection/management approaches (e.g., Storage Area Networks (SAN) and Network Attached Storage (NAS) [SNIA, DERRS, BARRC], iSCSI [DUPLS] and DAFS [DAFS]) have been promoted to address (at least some of) the associated storage management concerns. Regardless, the goal of common, controlled access by heterogeneous computer servers for uniform “data sharing” continues to be especially problematic. While definitions vary (e.g., [LEEBEN, EVALG]), “storage virtualization” has been proposed to aid in resolving several storage management issues (for instance, interoperability and storage allocation) and perhaps serve as the basis for enabling genuine “data sharing”. Regarding the impact upon overall storage I/O performance, further qualification of the cost/access metric previously noted might include consideration of requisite storage management (both as regards cost and the impact upon access).

All told and simply stated, computer hardware keeps getting faster and the amount of storage keeps growing. As suggested above, performance is a key factor within both of these major trends. Performance is also a problematic issue, for as previously noted, improvements in the performance of storage I/O have not kept pace with improvements in storage capacity. Even more unfortunate is the realization that this failure to keep pace is even more pronounced in comparison to the faster computer hardware.

Figure 1 shows the enormous difference between the speeds with which typical computers can access data internally within the microprocessor in comparison to the time that it takes to access data that resides upon storage devices.² The figure illustrates the relative latency times (i.e., the period of time between when the request for data is initiated and the start of the actual data transfer). For comparison purposes, the latency time for an internal transfer within the microprocessor itself has been arbitrarily assigned a value of one minute. The illustration shows that whereas an internal transfer of data within the microprocessor would take figuratively one minute, the microprocessor would have to wait one and one-half hours for data to become available from memory (and likewise two years if the data had to come from a disk device, assuming average seek and ½ rotation times).

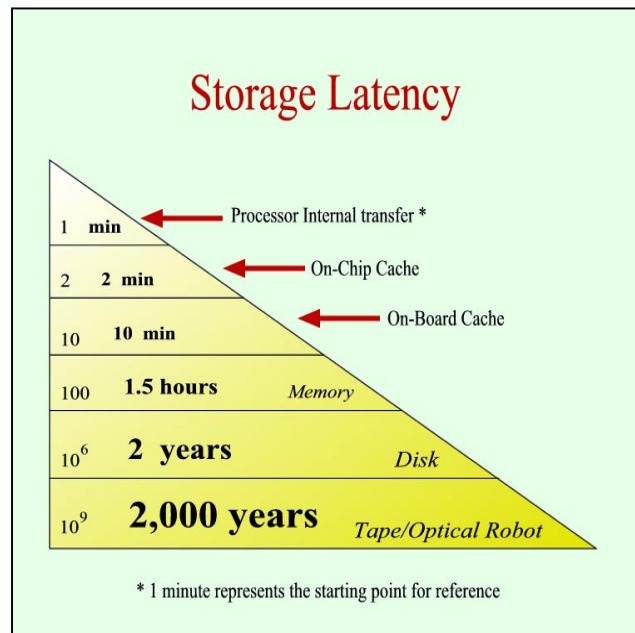


Figure 1 - Relative storage latencies

The gulf between the performance capabilities of storage and the performance capabilities of the faster computer hardware is chronic and has been observed for some time. For example, more than a decade ago this gulf was essentially termed an “I/O Crisis” where:

“Over the past decade, processing power, memory speed, memory capacity, and disk capacity have all grown tremendously: Single chip processors have increased in speed at the rate of 40%-100% per year; caches have increased in speed 40%-100% per year; main memory has quadrupled in capacity every two or three years. In contrast, disk access times have undergone only modest

² This figure is an adaptation of the “How far away is your data?” figure shown in [GRAY1]; another depiction comparing relative latencies (and bandwidths) can be found in [MASHEY].

performance improvements. For example, seek time has improved by only about 7% per year...If the imbalance is not remedied, Amdahl's law tells us that much of the astounding processor speedup and memory growth will be wasted." [Chen90]

This chronic performance gulf and the Amdahl/Case "balanced system law"³ have been discussed at length elsewhere (e.g., [GRAY2]). Besides increasing the base storage device performance characteristics (e.g., faster disk rotation speed, shorter disk seek times, etc.), additional attempts to remedy the chronic storage I/O operation performance gulf and achieve balanced systems have been diverse, sometimes novel and often complex. Examples include the introduction of disk RAID [PATSON], extensive and widespread caching techniques, increased parallelism, and many of the emerging technologies previously noted (such as NAS, SAN, DAFS, etc.) along with "storage virtualization" in general.

In sum, the two major trends (of faster computer hardware and the explosive growth of storage) will engender even greater attention and effort directed towards performance-related issues, and in particular the performance of storage I/O operations. Noteworthy here is that the ability to readily assess the actual storage I/O experience, especially in empirical terms, will be critical to the timely success of these efforts.

2.2 A Focus upon Files

Files live within this turbulent world of rapidly advancing computer hardware and increasing storage capacity. Within this realm, files play a principal role, since they not only support the applications that perform the tasks constituting the intended use of the computer, but they also enable the basic operation of the computer system itself. Much (if not most) of the actual work performed by computer systems generally requires significant file I/O operations. Moreover, as the momentum towards "storage virtualization" accelerates, files might play an even more expanded and pivotal role [MOOREF1].

At a more immediate level, files also represent the tangible, identifiable aspect of storage (in that files provide computer users and most application programs with the familiar interface and access to often dispersed data which resides upon the various storage devices). In common parlance, people and their applications deal with files, and rarely with devices directly per se. Files, in a sense, are the "end product" of storage. At least from an end-user perspective, issues raised and addressed in terms of (particularly specific) "files" generally have more immediate relevance and importance in comparison to talk of the underlying storage devices themselves. This stems from the observation that the nature of a file's use and the significance of its contents bring intrinsic meaning and importance to the file as a discrete entity. In this regard, files are the message; storage is just the medium.

Given their role, files are especially sensitive to and can be adversely impacted by performance-related factors and issues (such as the chronic storage I/O performance gulf previously discussed). By placing focus directly upon the I/O performance aspect of files, the extent of such impact can be better appreciated and understood in more immediate terms. Viewing and understanding the performance of storage I/O at the file level becomes even more important as files migrate towards the cloud of virtualized storage (or virtualized file systems for that matter). Assessing storage I/O performance at the file level becomes largely a practical necessity (or at the least, a first course of action) once a file is divorced from its direct connection to a specific physical device and the device's performance characteristics. In all, a more comprehensive and relevant assessment of storage I/O performance can be gained through the additional focus upon I/O performance specifically at the file level.

2.3 Measuring to meliorate and manage

The chronic gulf between the performance capabilities of storage and the performance capabilities of the

³ The Amdahl/Case "Balanced System Law" (or rule of thumb) basically states a balanced computer system needs 1 MB of main memory capacity and 1 Mbit per second of I/O bandwidth per MIPS of CPU performance. [AMDAHL]

faster computer hardware will likely become more visible when the introduction of faster computer hardware demonstrates little if any noticeable performance gains. This in turn makes expenditures on faster computer hardware more difficult to justify on an economic basis (especially since greater performance generally demands a premium price).

Given such a scenario, a key ability is that of monitoring the performance of storage I/O. This ability provides a cornerstone for better understanding the specific issues involved in the performance gulf dilemma, for developing and verifying potential solutions and further improvements, and for ensuring that implemented remedies continue to fulfill their objectives. In addition, granted that measurement is often a pre-requisite to management, better measurement capabilities enabled by monitoring can assist in addressing the current, immediate onus of storage management (which will become even more challenging given the ever-increasing growth of storage capacity). This “measure-to-manage” aspect will also gain in growing importance with the cost/access metric becoming a more dominant performance metric.

To date, the measuring and monitoring of storage I/O operation performance has for the most part been limited to and characterized by:

- 1) Hardware and software tools, basically of a diagnostic nature, that are used in the design and development of storage I/O-related products and/or to diagnose specific (ad hoc) system performance issues; these tools often provide knowledge-domain-specific functionality that is intended to satisfy a small audience (e.g., storage product developers, test lab engineers; etc.). Moreover, a number of these tools largely focus upon lower-level elements (for instance, data transfer sequences occurring on an I/O interface). All told, the hardware and software tools of this nature address a domain that is for the most part outside the need, usefulness, and interest of the typical computer (especially business) end-user.
- 2) Gross statistical counters that provide a limited level of granularity. For example, such counters might include the number of disk read operations per second, the number of bytes written to a particular disk, and perhaps at best the total number of read operations for all files. Unfortunately, these counters do not provide the full breadth and depth of required information about storage I/O operations (especially at the file level); moreover, it is often difficult if not impossible to correlate such counters with specific files (hence the I/O performance experience of particular files is indeterminate). Also susceptible to various ease-of-use deficiencies, these counters frequently prove to be seriously inadequate in assessing storage I/O performance at the file level, especially as regards to selected files of interest.
- 3) Benchmark programs that attempt to simulate actual customer applications. Certainly benchmarks provide a valuable service in that they promote a controlled, somewhat standardized method for deriving comparisons between alternative subjects under test; they might also be used to specifically target and stimulate a particular system element for investigation. However, benchmarks bear the burden of correlation to the actual-user system environment. That is, there is always the suspicion and argument as regards “how representative” a benchmark is and to what extent it corresponds to the actual operational environment of a particular computer end-user. While benchmarks can provide some helpful explanations, they often require a lot of explanation themselves.

A key new addition and complement to these current storage I/O performance monitoring capabilities would be the ability, in general practice, to adequately monitor the performance of storage I/O at the file-level. The rationale behind placing a focus specifically upon files has been discussed prior (q.v., section 2.2). The following sections discuss the benefits of file I/O operation performance monitoring and the associated operational/implementation requirements.

2.4 Benefits

The major benefits of monitoring the performance of storage I/O specifically at the file-level include:

- 1) Pertinence. The importance of relevance has been discussed in section 2.2. In comparison to the current storage I/O performance monitoring capabilities, file-level I/O performance information would, in many if not most cases, be more tangible, more relevant, more immediate and meaningful since this information would be directly related to the files themselves, a primary mechanism by which computer users and their applications generally interact with storage. Being able to easily and quickly (moreover, immediately) determine that a particularly important file is experiencing (or better yet, poised to experience) I/O performance problems (and to specifically what extent) can be a great advantage not only in performance/capacity management, but also in day-to-day operations.
- 2) Utility. As previously discussed (section 2.3), the ability to monitor storage I/O performance provides a cornerstone for addressing the issue of the chronic storage I/O performance gulf. It can help expedite the efforts of researchers, developers, and consultants in their attempts to investigate and propose remedies related to this issue (especially given a standard file I/O performance monitoring facility that is in common use and featured in actual production environments). As regards contemporary issues already confronted often on a daily basis, additional benefits include those associated with practical applications such as:
 - a) Fundamental performance analysis and tuning, and associated reduction in storage management costs. As the first step in analyzing performance problems with key applications, file I/O performance monitoring provides the ability to readily determine if the associated files are experiencing poor I/O performance (and to exactly what extent if so). With an automated file I/O performance monitor facility (especially one tied to an overall storage/system management facility), such efforts at performance analysis and tuning along with storage management efforts in general (e.g., determining and ensuring a more efficient and effective use of storage to meet the I/O performance needs of particular files) can be performed in a more timely and less resource-intense manner. Given the current shortage and cost of storage professionals, expedited efforts in this area are particularly welcome.
 - b) Determining the impact of system changes upon file I/O operation performance. File I/O performance monitoring affords the capability of determining the actual impact (with precise measurements as evidence) of changes within the computer system (such as the addition, replacement, or modification of disk subsystems, computer memory, file systems, processors, operating systems, etc.). Hence, it can help in the often-complex process of evaluating proposed improvements to computer systems (which can become especially difficult given the diversity of approaches sometimes proposed).
 - c) Product/system comparisons, particularly amongst vendors. Without having to rely only upon benchmarks (and the need to correlate how closely these benchmarks match one's particular application and corresponding computer system environment), file I/O performance monitoring allows one to use his/her actual applications and associated files in regular, normal operation with the products and/or systems that are the subject of comparison (such as different disk subsystems, computer servers, etc., perhaps from different vendors) so as to determine their effect upon (even specific) file I/O performance.
 - d) Monitor and confirm Quality of Service (QoS) for file I/O performance. File I/O performance monitoring provides the ability to (independently) monitor the actual I/O operation activity of specific files; such a capability would allow easier confirmation as to whether these files are meeting expectations as regards expected/contracted I/O performance.

e) Development and support aid. For a broad range of researchers, developers, consultants, and “troubleshooters” (including application developers concerned about file I/O performance; file system and device driver developers; disk subsystem developers; etc. – as well as the respective associated support personnel), a file I/O performance monitoring capability could provide a ready means of seeing exactly what (of interest) is happening in the file I/O activity (down to specific details for an individual file I/O operation).

3) Modest investment. Implemented in software alone, a file I/O performance monitoring facility would not require its user to assume any capital expenditures in hardware for its operation. In addition, implemented as an adjunct to the computer system, the file I/O performance monitoring facility can be relegated, if need be, to a benign role without disturbing normal system operation.

3. Basic Requirements for Monitoring File I/O Performance

This section of the paper delineates the basic requirements to be met by a file I/O performance monitoring facility. The following list of basic requirements is not meant to be mandatory in its entirety, given that some capabilities might only be considered as requirements due to the specific needs of a particular customer/situation/setting. In addition, the requirements are listed without any prioritization.

1) Ease-of-use. Ease-of-use is paramount if the file I/O performance monitoring capability is to be readily used (otherwise its use will be subject to resistance and aversion, no matter what its benefits). Ease-of-use generally entails a large element of subjectivity, but some basic, non-polemical requirements would include:

a) Flexibility and customization. Judicious options should be provided so that the user that easily configure the use of the facility and quickly focus in upon the particular items of interest.

b) Automation. Along similar lines, the facility should allow for ample automation so that the user can exploit the capabilities of the facility without excessive, repetitive manual effort.

c) Easy-in/Easy-out. The facility should allow for easy and quick installation, along with similar ease in de-installation if removal of the facility is desired.

2) Reliability. Given the critical role that files play within the computer system, the file I/O performance monitoring facility should not jeopardize the integrity of the computer system during its operation. In addition, the measurements and other output of the facility should be inerrant to the greatest extent possible.

3) Unintrusive with no system modifications required. The installation, operation and use of the facility should not require any of the operating system, file system, files, or application programs to be modified and/or recompiled.

4) Complete and pertinent information. The facility should capture and present complete and pertinent information about the file I/O operations that it is monitoring. At the minimum, the facility should support provision for response times, throughput (including data transfer rates and data transfer sizes), and I/O operation types and rates. The preceding statistical information should be selectively collected and be specific to a particular file of interest. In addition, the facility should allow options for collecting this information either on an individual file I/O operation basis and/or in summarized format. The summarized format option should further allow summaries on a periodic basis, at the conclusion (close) of file activity, or upon an exception basis. The exception basis should allow the user to set a specific threshold (e.g., a particular “I/O operations per second” rate) which when detected by the facility would cause the facility to produce a summary of the respective file’s I/O operations up to that point in time.

A Case for Monitoring File I/O Performance

The summarized file I/O operation performance information should also be available for use by other utility programs (for subsequent analysis, exception alerting, etc.).

- 5) Accuracy. The measurements taken by the file I/O performance monitoring facility should be consistently accurate and equate to corresponding values derived by other means that are known to be valid.
- 6) Precision. To be fully useful, the facility should be able to provide sufficient precision so that all file I/O operations can be observed with their respective timings recorded in adequate detail. Precision is particularly important with high-speed file I/O operations where such operations can be performed within well-under one millisecond. For example, a facility that provides precision only down to within 10 milliseconds will not suffice when file I/O operations can be performed in less than one millisecond.
- 7) Support both real-time and recorded modes of operation. The file I/O performance monitoring facility should be capable of recording and presenting statistical information about the performance of selected file I/O operations in real-time (i.e., as the file I/O operations occur). The facility should also optionally allow for the collected file I/O performance information to be recorded for subsequent, repetitive display and analysis.
- 8) Efficient. The component of the file I/O performance monitoring facility that is responsible for the actual observation of the file I/O operations and the collection of the respective performance information should be lightweight and efficient; it should incur the least amount of overhead as possible. For the most part, the minuscule amount of time required to perform the actual monitoring of the file I/O operations by the facility should represent a negligible addition to the overall time required to perform the file I/O operation itself (which can frequently be at least several milliseconds in duration). Furthermore, the facility option to allow a summarized, aggregated collection of file I/O performance information will help reduce the negligible overhead even further.
- 9) Platform independence and support. While it may support details about file I/O operation performance information that is specific to a particular platform (e.g., operating-system), the facility should provide a base set of statistical performance information (as noted above). This base set of statistical performance information (e.g., response times, throughput, etc.) should represent file I/O operation performance information that is essentially generic across platforms (and as such provides a common base for comparison across diverse platforms). As a corollary, the facility should support a variety of different platforms (e.g., operating systems, file systems, etc.).
- 10) Security. As part of its operation, the facility for file I/O performance monitoring should not access the actual data that is being transferred by means of the monitored file I/O operation. The facility should only reference at most the control structures related to the file I/O operation.

Implementations meeting the requirements above might vary widely depending upon the actual design; level of robustness, availability and serviceability; general development costs and pricing considerations; added features; etc.

4. Efforts at Implementing a File I/O Performance Monitor

For the most part, previous efforts at monitoring the performance of storage I/O at the file level have been limited to academic/research studies and but a few circumscribed commercial offerings. Such studies have often been limited to a particular operating system with the goal of probing a specific aspect of the computer system (e.g., file system design tradeoffs, characterization of file access activity, etc.). Furthermore, these studies often required “software instrumentation” (which necessitated that modifications be made to the application programs and/or other system components under observation). Such “software instrumentation” usually limited the studies to a small subset of application programs, with an additional limitation as to the

type and amount of the associated performance information collected. Examples of these academic studies and research efforts include the I/O extensions to the Pablo® instrumentation library [AYDT] as well as [PASQ] and [MATTHW].

One research study [VOGELS] avoided “software instrumentation” (as described above) and instead exploited an operating system mechanism that allows for “filtering” I/O operations. A software “trace agent” was developed as part of this study; this “trace agent” used the “I/O operation filtering” mechanism to intercept and collect performance information about file I/O operations. As a result, the study was able to glean fairly detailed performance information for a variety of files and their respective I/O operations.

In the commercial world, there are currently but a few products available that deal with monitoring files. The focus of most of these products is upon basic file storage management. That is, these products essentially deal with monitoring and reporting file attributes (e.g., the number and types of files; file sizes, locations and owners; file timestamps as regards creation date, last-access, etc.). Some provide additional functionality such as pro-active file quota management (which can be used to limit users as to the type of files they can access, the amount of disk storage they can use for files, etc.). File storage management products are offered by several vendors, including IBM Tivoli (www.ibm.com), EMC (www.emc.com), Sun Microsystems (www.sun.com), Symantec (Veritas) (www.symantec.com), NTP Software (www.ntpssoftware.com), and Tek-Tools (www.tek-tools.com).

While such file storage management products can certainly play an important role in overall storage management (and in potentially improving file I/O performance indirectly), they do not meet the requirements previously outlined for a file I/O performance monitoring facility (q.v. section 3). However, there are three currently available commercial products that provide some limited monitoring of actual file I/O operations:

- 1) IBM AIX® “filemon” command⁴. The IBM AIX (UNIX platform) operating system provides support for a “filemon” command; this command collects and presents trace data on the various layers of file system utilization specific to the AIX operating system. Using the system trace facility, the filemon command is able to collect a variety of performance information related to I/O activity during the elapsed time interval within which the command is running. The information collected about the I/O activity includes performance information related to the physical volumes, logical volumes, virtual memory segments, and active files. Both global and more detailed reports can be generated; these reports include various counters, with the detailed reports supplying specific performance information (such as the number of read and write I/O operations, response times, throughput, etc.) associated with a particular active file.

While the filemon command does provide a valuable set of performance information about file I/O operations, it is ostensibly designed and implemented essentially for diagnosing system performance issues (and is specific to the AIX system). It does not meet all of the requirements previously outlined for a file I/O performance monitoring facility (e.g., as regards the ability to select specific files of interest, provisions for the various summarized formats, support for exception thresholds, etc.).

- 2) FileSpy (Microsoft® Windows®). “FileSpy” is a general-purpose diagnostic tool that has been included, for example, within the Windows 2000 Resource Kit Tools package from Microsoft. This tool allows the user to monitor both local and network drives so as to observe the types of I/O Request Packets (IRP) and Fast I/O operations that are executing within the system. FileSpy is essentially a file system filter driver program that uses a console command-line interface. Logging information collected by the program can be displayed upon the console screen and/or directed to a specified file; this logging information is basically limited to diagnostic information such as sequence number, time stamps,

⁴ Additional documentation about the AIX “filemon” command can be found in [AIXFILEM].

operation type codes, process and thread IDs, operation completion status, File Object pointer and file name if available, etc.

Developed as an example for writing an Installable File System (IFS) filter driver under Windows 2000, FileSpy can provide useful diagnostic information for a limited audience (for example, software developers interested in observing some of the details underlying the file system activity within the system). However, like the AIX “filemon command”, FileSpy does not meet all of the requirements previously outlined for a file I/O performance monitoring facility (e.g., as regards the ability to select specific files of interest, provisions for the various summarized formats, support for exception thresholds, etc.).

- 3) Filemon for Windows NT/2000/9x.⁵ Filemon is a software program that monitors and displays file system activity (specifically performance information about file I/O operations) in real time. A graphical user interface (GUI) is used to select the specific files to be monitored and to display the file I/O operation information that has been collected. The file I/O operation information collected by Filemon includes the type of file I/O operation, a timestamp (or the time duration of the file I/O operation), the associated file name to which the file I/O operation was directed, the name of the process performing the file I/O operation, and the completion status associated with the file I/O operation.

Filemon provides a powerful diagnostic tool for observing file system I/O activity and obtaining I/O operation information about specific files of interest. Its GUI in particular helps promote ease-of-use. However, Filemon does not meet all of the requirements previously outlined for a file I/O performance monitoring facility (e.g., as regards provisions for the various summarized formats, throughput performance data, support for exception thresholds, etc.).

Designed to meet all of the requirements outlined above (q.v., section 3), the hIOmon architecture represents a file I/O operation performance monitoring facility that can examine and efficiently record the performance of user-selected file I/O operations for real-time and/or subsequent historical display.⁶ As shown in Figure 2, the hIOmon architecture is comprised of three (3) main components within a flexible and scalable architecture:

- 1) **Presentation Client.** The hIOmon “Presentation Client” is a Java application that provides a standard windows-type GUI. This GUI communicates (either locally or remotely over the network) with the hIOmon Manager to (initially) set the various control options of the hIOmon I/O Monitor (such as which particular files are to be monitored and the particular types of performance metrics to be collected); it can communicate with more than one hIOmon Manager concurrently. The Presentation Client can also be used to display the collected file I/O operation performance metrics (in real-time or “replay” mode) and optionally export the detailed and/or summarized file I/O operation performance metrics to a “Comma Separated Values (CSV)” file for use by various spreadsheet programs to perform customized analysis and/or to generate tailored graphs.
- 2) **Manager.** The hIOmon “Manager” runs as an operating-system service (or daemon). It manages the hIOmon I/O Monitor component (largely based upon control option requests from or set by the hIOmon Presentation Client); it also transforms, records, and transfers the collected file I/O operation performance metrics to the Presentation Client(s) and/or saves (concurrently) the performance

⁵ A freeware version of Filemon is available at the Microsoft “Windows Sysinternals” web site (technet.microsoft.com).

⁶ The hIOmon architecture has been implemented by hyperI/O LLC into a software product offering; the current version supports Microsoft Windows 2000/2003/2008 and Windows XP/Vista, provides an interface for use by Microsoft Windows Management Instrumentation (WMI), PerfMon/SysMon, and IBM Tivoli® Monitoring to display and monitor hIOmon-derived performance metrics for both files and disk devices, and is the only file (and disk) I/O operation performance monitoring facility currently available that satisfies all of the requirements described in section 3.

information to a specified disk file (for subsequent “replay” mode display). The hIOmon Manager can communicate with one or more Presentation Clients concurrently; it can also be configured to automatically start the hIOmon I/O Monitor component (so as to begin the actual monitoring of the selected files) when the Manager itself is started. The hIOmon Manager allows the hIOmon I/O Monitor to be much more simpler, faster and efficient in operation.

- 3) **I/O Monitor.** The hIOmon “I/O Monitor” is a lightweight component that performs the actual monitoring of the selected file (and disk) I/O operations; based upon such monitoring, it collects the requested file and disk I/O operation performance metrics. With an efficient design, the hIOmon I/O Monitor introduces very little overhead into the overall file or disk I/O operation (especially compared to the typical time durations of such I/O operations).

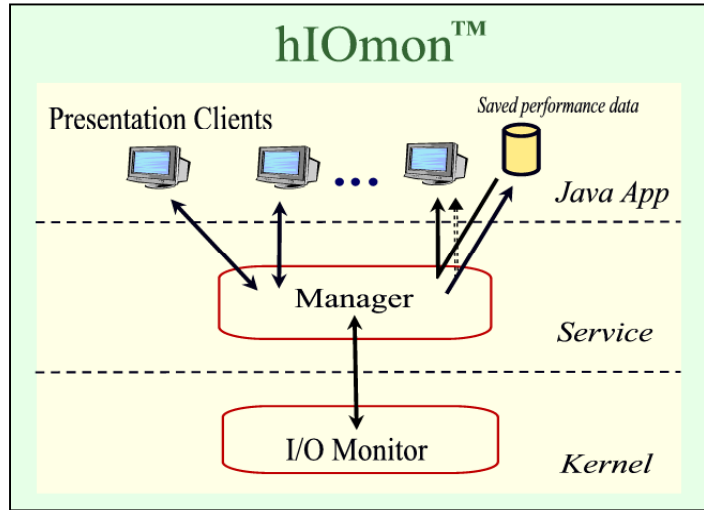


Figure 2 - hIOmon Architecture

The hIOmon Presentation Client provides the ability to select the particular files to be monitored by means of “filters”. Each of the selected file names (of files to be monitored) is considered to be a “filter”. That is, each file name represents a filter in that the hIOmon I/O Monitor will match the name against the file I/O operations it observes to determine whether or not the I/O operation is to be monitored (i.e., whether I/O operation performance metrics are to be collected for the I/O operation, and if so, the type of performance metric that is to be collected). Filters can be combined into groups called “Filter Selection” lists; this allows individual groups of “filters” to be defined (with each Filter Selection list representing a distinct group of files to be monitored).

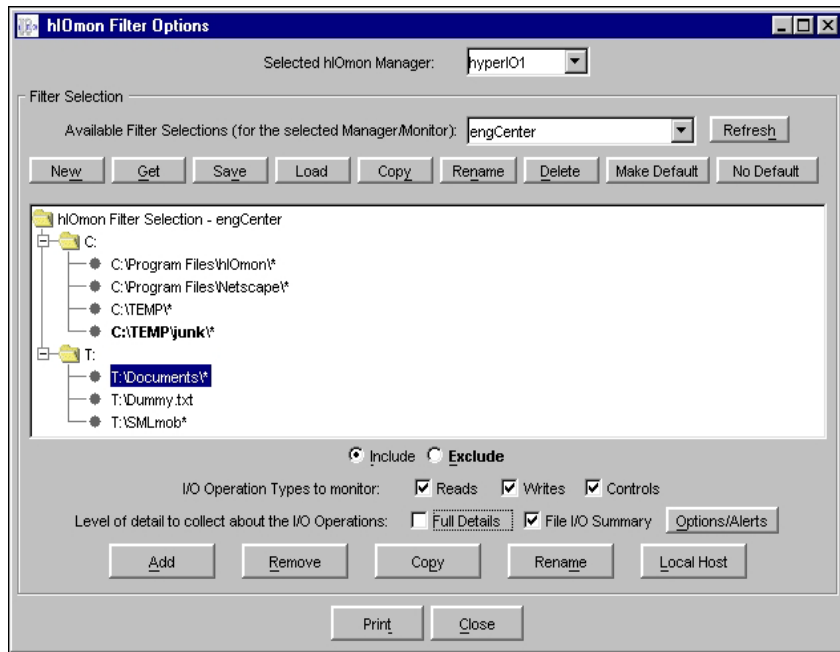


Figure 3 - hIOmon filter options example.

As shown in Figure 3, a number of options can be specified for each filter (file). For example, the filter can be set such that the hIOmon I/O Monitor will collect performance information for only read I/O operations, write I/O operations, and/or control I/O operations that are performed against the respective file(s). Filter options include the ability to request the collection of detailed (i.e., for each I/O operation) and/or summary performance metric information.

The File I/O summary option allows the user to specify that performance metrics for the respective file be collected on a periodic basis, when the file is closed, or upon an exception basis. A variety of thresholds

A Case for Monitoring File I/O Performance

(Alerts) are provided for summaries on an exception basis (as shown in Figure 4). These thresholds include average and maximum response times, amount of data transferred and transfer rates, and number of I/O operations performed; moreover, each of these thresholds (which can be set for a specific filter/file) can also be individually set for the particular I/O operation type (i.e., read, write or control). In addition, upon detection of these thresholds having been reached, the hIOMon Manager can optionally generate a System Event Log record (which in turn allows notification via system management tools that monitor the System Event Log). The hIOMon architecture also allows for hIOMon Manager extensions so as to support communication with alternative (e.g., customized) Clients as well as with overall storage/system management facilities (to provide for an integrated, comprehensive approach to system monitoring).

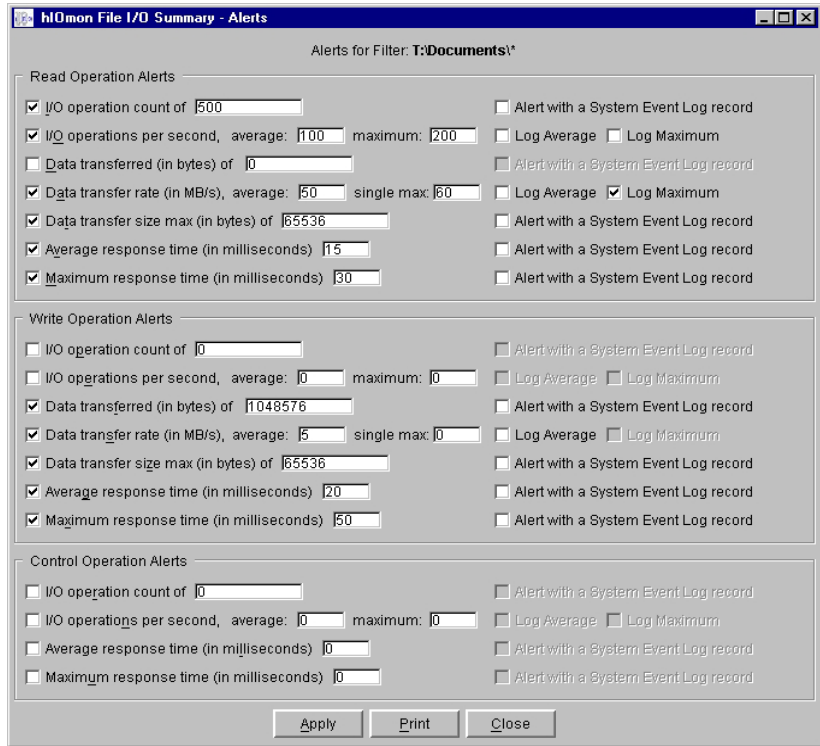


Figure 4 - hIOMon file filter Alert options

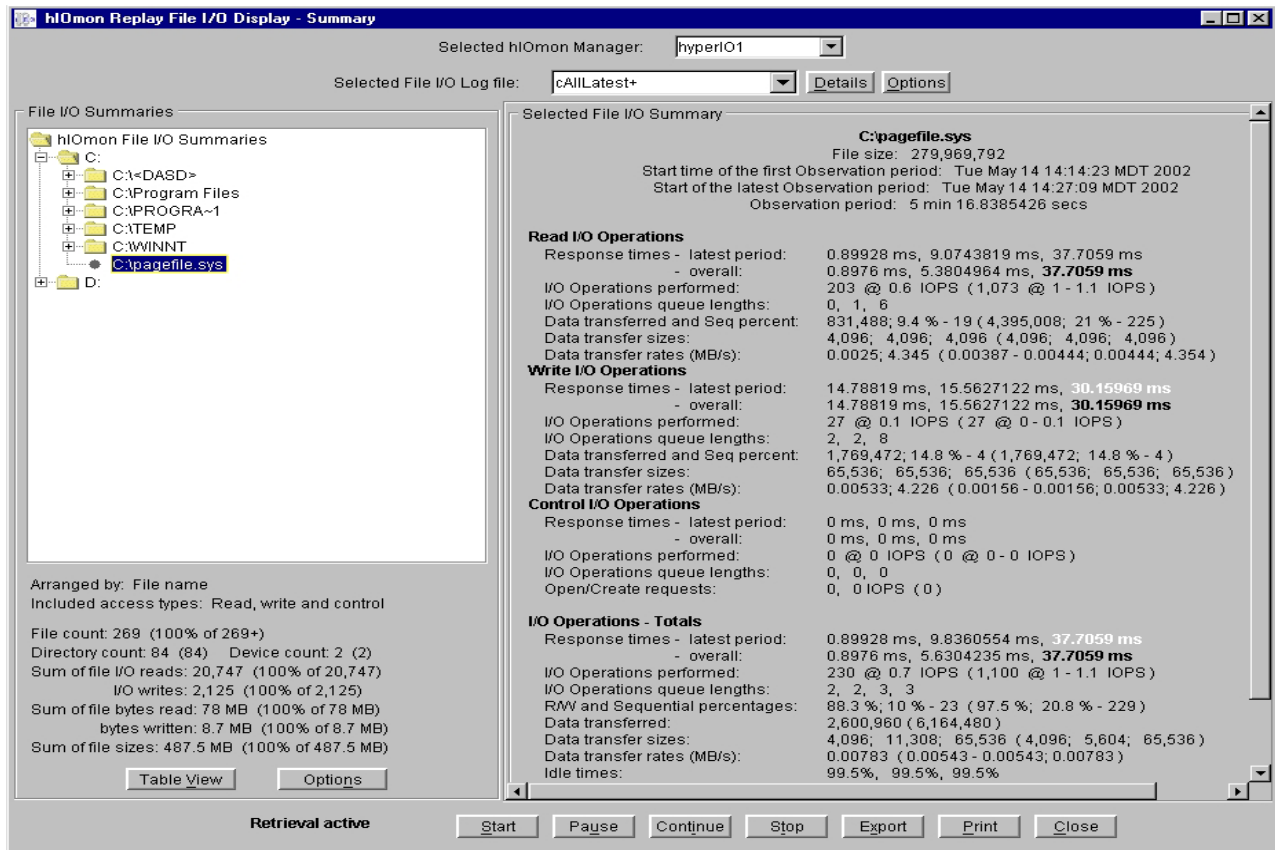


Figure 5 - hIOMon File I/O Summary display example.

Figure 5 shows an example of a “File I/O Summary” display provided by the hIOMon Presentation Client. Upon selecting the particular file of interest in the left-side pane, the file I/O performance summary information for the respective file is displayed within the right-side pane. A full set of performance information is provided in the display, including the observed number of I/O operations performed (per type); the minimum, average and maximum response times (with optional sub-millisecond precision); minimum, average, and maximum data transfer sizes; average and maximum data transfer rates, read/write ratios and the percentage of detected sequential access, along with the respective file size. Both current (for the most recent observation period) and cumulative (since the start of the first observation period) performance values are provided. In addition, performance values representing “alert thresholds” detected by the hIOMon I/O Monitor during the current observation period are highlighted in white text; bold text indicates that an “alert threshold” had been detected since the start of the initial observation period.

In sum, hIOMon represents the introduction of a file I/O performance monitoring facility that can successfully meet all of the requirements outlined in section 3. It provides the functionality required to easily, quickly and adequately monitor the I/O operation performance of selected files (and associated disk devices as well). The hIOMon architecture also provides the necessary extensibility to support adaptations required to satisfy more customized monitoring needs.

5. Conclusion

Faster computer hardware and more disk storage – a combination of trends that is obvious to even the casual observer. But perhaps less obvious is the chronic performance gulf between this fast computer hardware and the storage upon which computer systems so heavily depend. Unfortunately, the ongoing advance of faster computer hardware together with the explosive growth in storage has further aggravated this performance dilemma. If the potential performance and ensuing productivity increases afforded by faster computer hardware are to be fully and profitably realized, then this performance gulf must be successfully addressed (or at least, better mitigated). The ability to adequately monitor storage I/O performance will be one of the key competencies required in efforts meant to address the performance gulf and to better manage the explosive growth of storage in general. Given the crucial role that files play both within current computer systems and within the emerging world of “virtualization” (both storage and systems), such efforts will more likely and quickly succeed given the ability to readily measure and monitor the performance of file I/O operations in particular. A file I/O performance monitoring facility, such as one based upon the hIOMon architecture, provides a beginning basis by which one can readily and reasonably respond, with empirical metrics in hand, to the increasingly-important question, “*How fast are your files?*”

6. References

- [10GBE] 10 Gigabit Ethernet Alliance, <http://www.10gea.org/>
- [AIXFILEM] “RS/6000 Performance Tools in Focus”, SG24-4989-00, *IBM International Technical Support Organization – Austin Center*, May 1997
- [AMDAHL] Amdahl, Gene, “Storage and I/O Parameters and System Potential”, *IEEE Computer Group Conference*, January 1970, pp. 371-72
- [AYDT] Aydt, Ruth A., “A User’s Guide to Pablo® I/O Instrumentation”, *University of Illinois*, December 1994 (Revised October 1996), <http://www-pablo.cs.uiuc.edu/Publications/Documents/documents.htm>
- [BARRC] Barrera, Clodoaldo, “Technical Directions for Storage Networking”, *Storage Networking Industry Association presentation*, Fall Comdex 2000, http://www.snia.org/English/Collaterals/Presentations/20001113_Technology_Directions_Barrera.pdf
- [BRECHR] Brechtlein, Richard, “Moving Toward Storage Re-Centralization”, *InfoStor*, January 2001, Volume: 5 Issue: 1, <http://www.infostor.com/>
- [CHEN90] Chen, Peter M., Garth A. Gibson, Randy H. Katz and David A. Patterson, “An Evaluation of Redundant Arrays of Disks using an Amdahl 5890”, *Proceedings of the ACM Conference on Measurement and Modeling of Computer Systems*, May 1990, Boulder, Colorado, pp. 74-85

A Case for Monitoring File I/O Performance

- [DAFS] Direct Access File System (DAFS) Collaborative, <http://www.dafscollaborative.org/>
- [DERRS] Derrington, Sean (META Group), “NAS and SAN Storage: Separate but Equal – Part 1” and “NAS and SAN Storage: Separate but Equal – Part 2”, *EMC Industry Analyst Reports*, <http://www.emc.com/news/analyst/od839.pdf>, <http://www.emc.com/news/analyst/od838.pdf>
- [DUPLS] Duplessie, Steve, “Block-level Storage over IP”, *InfoStor*, January 2001, Volume: 5 Issue: 1, pp. 64-67.
- [EVALG] Evaluator Group, Inc., “Virtualization of Disk Storage”, WP-0007-1, September 2000, page 1.
- [FCIA] Fibre Channel Industry Association, <http://www.fibrechannel.org/>
- [GILDER] Gilder, George, “Fiber Keeps Its Promise”, *Forbes*, 7 April, 1997, http://www.gildertech.com/public/telecosm_series/promise.html
- [GRAY1] Gray, Jim, “Computer Technology Forecast for Virtual Observatories”, *Microsoft Technical Report MSR-TR-2000-102*, September 2000, http://research.microsoft.com/~gray/papers/MSR_TR_102_VOF_Technology_Forecast.PDF
- [GRAY2] Gray, Jim and Prashant Shenoy, “Rules of Thumb in Data Engineering”, *Microsoft Technical Report MSR-TR-99-100*, Dec 1999 (Revised March 2000), http://research.microsoft.com/~gray/papers/MS_TR_99_100_Rules_of_Thumb_in_Data_Engineering.pdf
- [GRAY3] Gray, Jim and Goetz Graefe, “The Five Minute Rule Ten Years Later, and Other Computer Storage Rules of Thumb”, http://research.microsoft.com/~gray/5_min_rule_SIGMOD.doc
- [INFTRADE] InfiniBand(sm) Trade Association, <http://www.infinibandta.org/>
- [LEEBEN] Lee, Richard E. and Harriett L. Bennett, “SANs Rely On Storage Virtualization”, *InfoStor*, January 2001, Volume: 5 Issue: 1, pp. 20-28.
- [LYMAN] Lyman, Peter, Hal R. Varian, James Dunn, Aleksey Strygin, and Kirsten Swearingen, “How Much Information?”, *School of Information Management and Systems*, University of California at Berkeley, <http://www.sims.berkeley.edu/how-much-info/index.html>
- [MASHEY] Mashey, John R., “Big Data and the Next Wave of InfraStress: Problems, Solutions, Opportunities”, *1999 USENIX Annual Technical Conference*, June 1999, http://www.usenix.org/events/usenix99/invited_talks/mashey.pdf
- [MATTHW] Matthews, Jeanna Neefe, Drew Roselli, Adam M. Costello, Randolph Y. Wang and Thomas E. Anderson, “Improving the Performance of Log-Structured File Systems with Adaptive Methods”, *Proceedings of the 16th ACM Symposium on Operating Systems Principles*, October 1997, Saint Malo France, pp. 238-251
- [MOOREF1] Moore, Fred, “Without a Revolutionary File Management Solution There Can Be No True SAN”, *Computer Technology Review*, September 2000
- [MOOREF2] Moore, Fred, “Digital Data’s Future – You Ain’t Seen Nothin’ Yet”, *Computer Technology Review*, October 2000
- [MOOREG] Moore, Gordon E., “Moore’s Law”, *Fall 1997 Intel Developer Forum*, October 1997, <http://developer.intel.com/update/archive/issue2/feature.htm>
- [PASQ] Pasquale, Barbara K. and George C. Polyzos, “Dynamic I/O Characterization of I/O Intensive Scientific Applications”, *Proceedings of the Conference on Supercomputing 1994*, November 1994, Washington, U.S.A., pp. 660-669
- [PATSON] Patterson, David A., Garth A. Gibson and R. Katz, “A Case for Redundant Array of Inexpensive Disks (RAID)”, *ACM SIGMOD Conference Proceedings*, June 1988, Chicago, Illinois, pp. 109-116
- [PCI-X] Compaq, “PCI-X Enablement Program”, <http://www5.compaq.com/products/servers/technology/pci-x-enablement.html>
- [SCSI-7] SCSI Trade Association (STA), “The Seven Generations of SCSI”, <http://www.scscita.org/aboutscsi/7gen.html>
- [SNIA] Storage Networking Industry Association, <http://www.snia.org/>
- [VOGELS] Vogels, Werner, “File system usage in Windows NT 4.0”, *Proceedings of the 17th ACM Symposium on Operating Systems Principles*, December 1999, Charleston U.S.A., pp. 93-109
- [WHITING] Whiting, Rich, “Survey: Very Large Databases Keep Getting Bigger”, *Information Week*, 29 January 2001, <http://www.informationweek.com/822/ibm.htm>