

Deducing the Effects of File System Performance on System Reliability

By Michael Materie

Introduction

A computer system, like any other system, is only as fast as its slowest link. On a computer, the weakest link is typically the hard drive, a mechanical device in an environment of electronic operations. Accessing a spinning magnetic device through disk heads (operations measured in milliseconds) is much slower than the electronic communications that take place between a CPU and memory (nanoseconds). A slow disk subsystem can cripple the fastest combinations of CPUs and memory. Applications, and the operating system itself, interact with the disk subsystem via a structured set of layers. This ensures a symbiosis between a file system driver (e.g. ntfs.sys) and the disk subsystem. Bottlenecks anywhere in this layered structure reflect on the overall performance. When this article describes file system performance, it relates to both file system level considerations as well as disk subsystem considerations.

Over the years, numerous manufacturers, third-party analysts and labs have reported on the effects of file systems on system speed and performance, but information on the tangential effects of file systems on computer system reliability and stability, seemingly, remains very much a mystery.

The sheer magnitude of variables makes problem diagnosis a massively complex task for IT professionals when something goes wrong. This article will help to uncover some of the more common reliability and downtime phenomena associated with file system performance. My hope is that the data here will help you diagnose issues, and put you on the path to implementing the proper permanent solutions.

An Overview of the Problem

The principle of a slow file system's impact on system or application reliability is the timing out of a requestor or service provider in collecting/reassembling data. Imagine, as an example of physical limitations, a critical production file server hosting hundreds of users' data on a single IDE disk. No level-headed IT professional would consider this a workable solution. Needless to say, that configuration would never work; the I/O requests would be far greater than the ability to service those requests.

In compliment to I/O load balancing, administrators should include file system level solutions. Such solutions are unlikely to replace the need for I/O load disbursement, but they certainly compliment it, and in some cases *can* make an unworkable situation viable.

One of the key "logical" file system considerations is storing data files in contiguous manner on the hard drive(s). This is a key factor in

keeping the file system performing at peak efficiency. Though unavoidable, the moment a file is broken into pieces and scattered across a drive, it opens the door to the host of stability/reliability issues that inappropriate/inefficient disk subsystem would. Having just a few key files heavily fragmented can lead to crashes, conflicts and errors in the same manner that the single IDE disk server example would.

The principle of fragmentation-related issues holds true for both IP datagram fragmentation and file/disk fragmentation. Many system and application breakage points can be defined as "exerted stress on buffers to the point of overflow/overrun." DoS attacks are well-documented examples of exploiting IP datagrams, but far less information abounds for reliability considerations in the case of file objects. A good overview of the affect of stress when requesting file objects comes from a Microsoft® Knowledge Base article which states "The Server service cannot process the requested network I/O items to the hard disk quickly enough to prevent the Server service from running out of resources."

Disk fragmentation is often the "straw that broke the camel's back" when noting issues of stability or reliability.

Disk fragmentation is often the "straw that broke the camel's back" when noting issues of stability or reliability. Stressed I/O activity, compounded by fragmentation, can expose faulty device drivers or file filters that may otherwise operate effectively (in non-fragmented environments). The reliability of third-party applications is highly dependent on the degree to which those applications can accommodate bottlenecks, such as in disk subsystems.

The point at which application or system stability is compromised is difficult, if not impossible, to calculate. It is a combination of hardware and software and operations at the moment of instability. A poorly written driver or file filter can be exposed in some environments but not in others, and the amount of fragmentation required to reach "critical mass" on a specific file or files will vary greatly upon all the other variables involved.

This issue can be exemplified by better understanding asynchronous I/O. Asynchronous I/O exists to compensate for variables that may prevent or eliminate the possibility of synchronous I/O (e.g. I/O is much slower than data processing). The alternative to handling I/O

asynchronously, which generally offers lower performance, is to “block” other I/O.

Example: The developer of an application (Win32) can create either an I/O completion port, execute an overlapping completion routine, or call `WaitForSingleObject / WaitForMultipleObjects` APIs at the time of thread creation. In any case where the wait state is exceeded (e.g. queued I/O is paged to disk), a failure can occur. Low available memory (non-paged pool) can exacerbate failures as it re-introduces the physical disk into the equation.

In lieu of these failures, the developer can extend queuing/waiting and implement exception handling to mitigate issues, at the expense of lower performance (operations take longer) for the application, or an increase in system resource requirements.

Tracing Reliability Issues Via Errors and Performance Monitoring

There are many documented cases of errors and crashes on Windows and third-party applications caused by impacted file system performance, many solved by defragmentation. These types of errors include but are not limited to system hangs, time outs, failure to load, failure to save data and in worse case blue screens (where fragmentation aggravates flawed device drivers).

Perhaps the most prevalent of these circumstances in modern systems is the Event ID 2021 and 2022 errors found on systems hosting data. See figure 1.

```
Event ID: 2021
Source: Srv
Description: Server was unable to create a work item n times in the last seconds seconds.
Event ID: 2022
Source: Srv
Description: Server was unable to find a free connection n times in the last seconds seconds.
```

Figure 1: Event ID 2021 and 2022 errors

```
Event ID: 3013
Source: Rdr
Description: The redirector has timed out to computer name.
Status code 1450: Insufficient system resources exist to complete the requested service.
```

Figure 2: The client requesting the data will return related errors along the lines of Event ID 3013 or status code 1450.

An MS TechNet article from the Microsoft Windows 2000 Professional Resource Kit in Chapter 30 “Examining and Tuning Disk Performance” notes defragmentation as a primary solution to resolving disk bottlenecks such as those identified by the above detailed Physical Disk counters.

According to Microsoft Support article 822219, “You experience slow file server performance and delays occur when you work with files that are located on a file server.” It notes: “Use Performance Logs and Alerts to monitor the Avg. Disk Queue Length counter of the PhysicalDisk performance object.” Next is a list of symptoms quoted from that article:

- ▼ *A Windows-based file server that is configured as a file and print server stops responding and file and print server functionality temporarily stops.*

It is important to note that in a corporate IP network bottlenecks may be incorrectly advertised or diagnosed as network-related bottlenecks. In reality these bottlenecks often exist in the disk subsystem on a remote system.

In such circumstance the client requesting the data will return related errors along the lines of Event ID 3013 or status code 1450. See figure 2.

It is important to note that in a corporate IP network bottlenecks may be incorrectly advertised or diagnosed as network-related bottlenecks. In reality these bottlenecks often exist in the disk subsystem on a remote system. The specification of Windows file sharing services (CIFS) is such that file requests (supposedly only “valid” ones) will time out as the reliability of the network is a variable that might otherwise cause undue and unnecessary wait requests (should a client be disconnected). In reality extended waits can be interpreted as dropped client connections.

An important clue to investigating fragmentation as a potential or lead contributor to Reliability-I/O issues are when recommendations are made (by a support article or support engineer) to measure the following Physical Disk Counters related to Disk I/O:


- ▼ Average Disk Queue Length
- ▼ Average Disk Read Queue Length
- ▼ Average Disk Write Queue Length
- ▼ Average Disk Sec/Read
- ▼ Average Disk Sec/Transfer
- ▼ Average Disk Writes/Sec
- ▼ Split I/Os

- ▼ *You experience an unexpectedly long delay when you open, save, close, delete, or print files that are located on a shared resource.*
- ▼ *You experience a temporary decrease in performance when you use a program over the network. Performance typically slows down for approximately 40 to 45 seconds. However, some delays may last up to 5 minutes.*
- ▼ *You experience a delay when you perform file copy or backup operations.*
- ▼ *Windows Explorer stops responding when you connect to a shared resource or you see a red X on the connected network drive in Windows Explorer.*
- ▼ *You receive an error message similar to one of the following messages when you try to connect to a shared resource:*
 - Error message 1*
System error 53. The network path was not found.
 - Error message 2*
System error 64. The specified network name is no longer available.
- ▼ *You are intermittently disconnected from network resources, and you cannot reconnect to the network resources on the file server. However, you can ping the server, and you can use a Terminal Services session to connect to the server.*

- ▼ *If multiple users try to access Microsoft Office documents on the server, the “File is locked for editing” dialog box does not always appear when the second user opens the file.*
- ▼ *A network trace indicates a 30- to 40-second delay between an SMB Service client command and a response from the file server.*
- ▼ *When you try to open an Access 2.0 database file (.mdb file) in Microsoft Access 97, in Microsoft Access 2000, or in Microsoft Access 2002, you may receive an error message that is similar to the following:
Disk or network error.*
- ▼ *When you try to open a Microsoft Word file, you may receive the following error message:
Word failed reading from this file file_name. Please restore the network connection or replace the floppy disk and retry.*
- ▼ *When you log on to the file server, after you type your name and password in the Log On to Windows dialog box, a blank screen appears. The desktop does not appear.*
- ▼ *A program that uses remote procedure call (RPC) or uses named pipes to connect to a file server stops responding.*

Conclusion

Resolving issues with file system performance include numerous solutions, primarily hardware related, from use of faster disks, a greater number of disks, distributed storage, SANs, and on the bleeding edge extreme, petabyte-worthy technologies such as cluster file systems. It can also lead to “workaround” handlings such as reinstalling software, re-imaging of hard drives, replacement of hardware, all of which incur overwork on the administrative end. It forces IT to work reactively on problems, increasing IT costs and adversely affects user productivity due to unacceptable levels of downtime.

For all these solutions and workarounds, none negate the negative impact of fragmentation. Therefore a file system performance utility should, minimally, be evaluated alongside, and even sometimes in lieu of, other options. 

NaSPA member Michael Materie (pm@diskeeper.com) is the Director of Product Management at Diskeeper Corporation. He is a Microsoft Certified System Engineer (MCSE) and a Cisco Certified Network Associate (CCNA), and is A+ and I-Net+ certified.