Taking Your Server's Pulse

Steven Choy

AT A GLANCE:

- Customizing Performance Monitor
- Guidance on what and how often to measure
- An overview of key counters and what to look for

An overview of key counters and what to look for

Contents

Imagine you just arrived at the office on a Monday morning and you're greeted by an eager user who is complaining that his server is running too slow. How do you even begin to help him?

Performance Monitor, a handy tool built into Windows®, can assist you in diagnosing the problem. You can access Performance Monitor by typing perfmon at the command prompt or by selecting the Performance or Reliability and Performance Monitor (in Windows Vista® and Windows Server® 2008) from the Administrative Tools menu. To add performance counters and objects to be monitored, you simply click the plus sign and select from a host of possible choices.

So how do you measure the pulse of a server? There are more than 60 basic performance objects, and each object contains multiple counters. In this article, I will discuss the counters that reveal the vital signs of a server, and I will describe the typical sampling intervals that Microsoft® Service Support engineers use most often to troubleshoot performance-related issues.

Of course, a baseline provides a critical reference point when troubleshooting. Since the server load depends on the business requirements and also varies from time to time depending on the business

cycle, it is important to establish a baseline determined by the normal workload over a specified period of time. That allows you to observe changes and identify trends.

Making the Results More Readable

Before I dive into an analysis of the counters that represent the vital signs of servers, I'll tell you about two tricks that will make it easier for you to measure the vital signs of servers using Performance Monitor. Note that these tricks are not needed in Windows Vista and Windows Server 2008, but if you are running Performance Monitor on earlier versions of Windows, these two tweaks can come in very handy.

First, you can remove all the distracting sample noise that obscures the graphical view of trend lines. In Windows Vista and Windows Server 2008, Performance Monitor can display up to 1000 data points in graphical view. In previous versions of Windows, the limit is only 100 data points. When there are more than 100 points, Performance Monitor "buckets" the data points. A bucket is represented by a vertical line, indicating the minimum, average, and maximum of the sample points included in the bucket.

As you can tell by looking at the graph in **Figure 1**, it is difficult to spot the trend line when so much data is displayed at the same time. The **Figure 2** graph shows how much easier it is to grasp the data quickly when all the extraneous visual information has been turned off. For details on how you can turn off these vertical lines, see the Knowledge Base article that is available at support.microsoft.com/kb/283110.

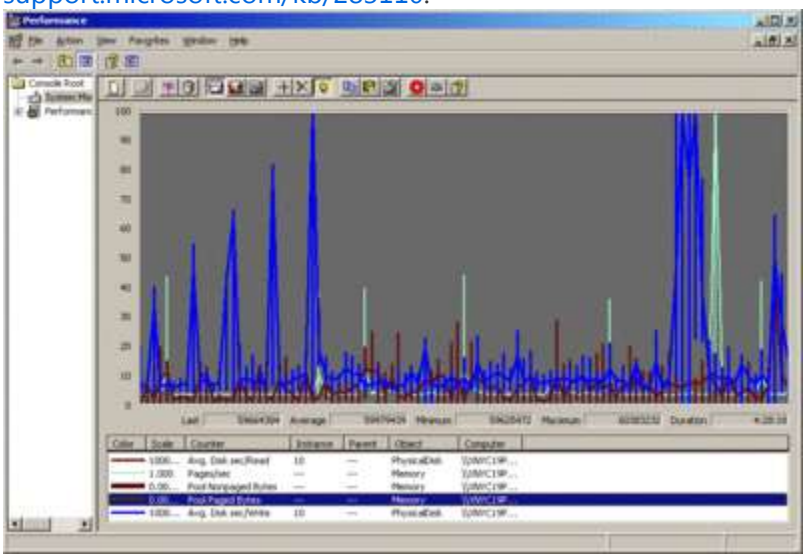

Figure 1 **Performance data shown with distracting buckets and no commas** (Click the image for a larger view)

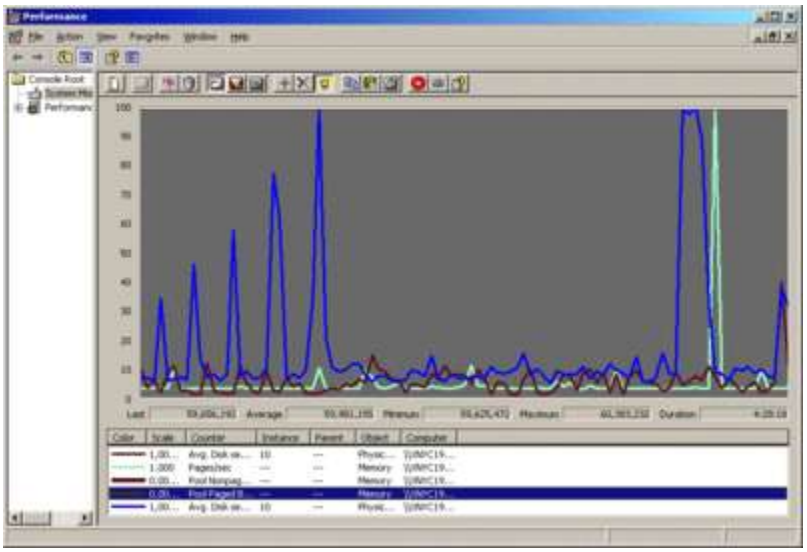

Figure 2 **A cleaner view of data with comma separators** (Click the image for a larger view)

The second trick is to add comma separators in the numbers, making it much easier to read the values shown in the counters. Windows Vista and Windows Server 2008 have comma separators enabled by default. In previous versions of Windows, however, Performance Monitor does not enable commas by default.

This may not sound like it would make a huge difference, but take a look at **Figure 1**, which shows the performance counters without commas, and then look at **Figure 2**, which shows the counters with commas. I find the latter much more readable. For some simple instructions on adding comma separators to your performance counters in Windows XP, take a look at the Knowledge Base article at support.microsoft.com/kb/300884.

What and When to Measure

Bottlenecks occur when a resource reaches its capacity, causing the performance of the entire system to slow down. Bottlenecks are typically caused by insufficient or misconfigured resources, malfunctioning components, and incorrect requests for resources by a program.

There are five major resource areas that can cause bottlenecks and affect server performance: physical disk, memory, process, CPU, and network. If any of these resources are overutilized, your server or application can become noticeably slow or can even crash. I will go through each of these five areas, giving guidance on the counters you should be using and offering suggested thresholds to measure the pulse of your servers.

Since the sampling interval has a significant impact on the size of the log file and the server load, you should set the sample interval based on the average elapsed time for the issue to occur so you can establish a baseline before the issue occurs again. This will allow you to spot any trend leading to the issue.

Fifteen minutes will provide a good window for establishing a baseline during normal operations. Set the sample interval to 15 seconds if the average elapsed time for the issue to occur is about four hours. If the time for the issue to occur is eight hours or more, set the sampling interval to no less than five minutes; otherwise, you will end up with a very large log file, making it more difficult to analyze the data.

Hard Disk Bottleneck

Since the disk system stores and handles programs and data on the server, a bottleneck affecting disk usage and speed will have a big impact on the server's overall performance.

Please note that if the disk objects have not been enabled on your server, you need to use the command-line tool Diskperf to enable them. Also, note that % Disk Time can exceed 100 percent and, therefore, I prefer to use % Idle Time, Avg. Disk sec/Read, and Avg. Disk sec/write to give me a more accurate picture of how busy the hard disk is. You can find more on % Disk Time in the Knowledge Base article available at support.microsoft.com/kb/310067.

Following are the counters the Microsoft Service Support engineers rely on for disk monitoring.

LogicalDisk\% Free Space This measures the percentage of free space on the selected logical disk drive. Take note if this falls below 15 percent, as you risk running out of free space for the OS to store critical files. One obvious solution here is to add more disk space.

PhysicalDisk\% Idle Time This measures the percentage of time the disk was idle during the sample interval. If this counter falls below 20 percent, the disk system is saturated. You may consider replacing the current disk system with a faster disk system.

PhysicalDisk\Avg. Disk Sec/Read This measures the average time, in seconds, to read data from the disk. If the number is larger than 25 milliseconds (ms), that means the disk system is experiencing latency when reading from the disk. For mission-critical servers hosting SQL Server® and Exchange Server, the acceptable threshold is much lower, approximately 10 ms. The most logical solution here is to replace the current disk system with a faster disk system.

PhysicalDisk\Avg. Disk Sec/Write This measures the average time, in seconds, it takes to write data to the disk. If the number is larger than 25 ms, the disk system experiences latency when writing to the disk. For mission-critical servers hosting SQL Server and Exchange Server, the acceptable threshold is much lower, approximately 10 ms. The likely solution here is to replace the disk system with a faster disk system.

PhysicalDisk\Avg. Disk Queue Length This indicates how many I/O operations are waiting for the hard drive to become available. If the value here is larger than the two times the number of spindles, that means the disk itself may be the bottleneck.

Memory\Cache Bytes This indicates the amount of memory being used for the file system cache. There may be a disk bottleneck if this value is greater than 300MB.

Memory Bottleneck

A memory shortage is typically due to insufficient RAM, a memory leak, or a memory switch placed inside the boot.ini. Before I get into memory counters, I should discuss the /3GB switch.
More memory reduces disk I/O activity and, in turn, improves application performance. The /3GB switch was introduced in Windows NT® as a way to provide more memory for the user-mode programs.

Windows uses a virtual address space of 4GB (independent of how much physical RAM the system has). By default, the lower 2GB are reserved for user-mode programs and the upper 2GB are reserved for kernel-mode programs. With the /3GB switch, 3GB are given to user-mode processes. This, of course, comes at the expense of the kernel memory, which will have only 1GB of virtual address space. This can cause problems because Pool Non-Paged Bytes, Pool Paged Bytes, Free System Page Tables Entries, and desktop heap are all squeezed together within this 1GB space. Therefore, the /3GB switch should only be used after thorough testing has been done in your environment.
This is a consideration if you suspect you are experiencing a memory-related bottleneck. If the /3GB switch is not the cause of the problems, you can use these counters for diagnosing a potential memory bottleneck.

Memory\% Committed Bytes in Use This measures the ratio of Committed Bytes to the Commit Limit—in other words, the amount of virtual memory in use. This indicates insufficient memory if the number is greater than 80 percent. The obvious solution for this is to add more memory.

Memory\Available Mbytes This measures the amount of physical memory, in megabytes, available for running processes. If this value is less than 5 percent of the total physical RAM, that means there is insufficient memory, and that can increase paging activity. To resolve this problem, you should simply add more memory.

Memory\Free System Page Table Entries This indicates the number of page table entries not currently in use by the system. If the number is less than 5,000, there may well be a memory leak.

Memory\Pool Non-Paged Bytes This measures the size, in bytes, of the non-paged pool. This is an area of system memory for objects that cannot be written to disk but instead must remain in physical memory as long as they are allocated. There is a possible memory leak if the value is greater than 175MB (or 100MB with the /3GB switch). A typical Event ID 2019 is recorded in the system event log.

Memory\Pool Paged Bytes This measures the size, in bytes, of the paged pool. This is an area of system memory used for objects that can be written to disk when they are not being used. There may be a memory leak if this value is greater than 250MB (or 170MB with the /3GB switch). A typical Event ID 2020 is recorded in the system event log.

Memory\Pages per Second This measures the rate at which pages are read from or written to disk to resolve hard page faults. If the value is greater than 1,000, as a result of excessive paging, there may be a memory leak.

Processor Bottleneck

An overwhelmed processor can be due to the processor itself not offering enough power or it can be due to an inefficient application. You must double-check whether the processor spends a lot of time in paging as a result of insufficient physical memory. When investigating a potential processor bottleneck, the Microsoft Service Support engineers use the following counters.

Processor\% Processor Time This measures the percentage of elapsed time the processor spends executing a non-idle thread. If the percentage is greater than 85 percent, the processor is overwhelmed and the server may require a faster processor.

Processor\% User Time This measures the percentage of elapsed time the processor spends in user mode. If this value is high, the server is busy with the application. One possible solution here is to optimize the application that is using up the processor resources.

Processor\% Interrupt Time This measures the time the processor spends receiving and servicing hardware interruptions during specific sample intervals. This counter indicates a possible hardware issue if the value is greater than 15 percent.

System\Processor Queue Length This indicates the number of threads in the processor queue. The server doesn't have enough processor power if the value is more than two times the number of CPUs for an extended period of time.

Network Bottleneck

A network bottleneck, of course, affects the server's ability to send and receive data across the network. It can be an issue with the network card on the server, or perhaps the network is saturated and needs to be segmented. You can use the following counters to diagnosis potential network bottlenecks.

Network Interface\Bytes Total/Sec This measures the rate at which bytes are sent and received over each network adapter, including framing characters. The network is saturated if you discover that more than 70 percent of the interface is consumed. For a 100-Mbps NIC, the interface consumed is 8.7MB/sec (100Mbps = 100000kbps = 12.5MB/sec* 70 percent). In a situation like this, you may want to add a faster network card or segment the network.

Network Interface\Output Queue Length This measures the length of the output packet queue, in packets. There is network saturation if the value is more than 2. You can address this problem by adding a faster network card or segmenting the network.

Process Bottleneck

Server performance will be significantly affected if you have a misbehaving process or non-optimized processes. Thread and handle leaks will eventually bring down a server, and excessive processor

usage will bring a server to a crawl. The following counters are indispensable when diagnosing process-related bottlenecks.

Process\Handle Count This measures the total number of handles that are currently open by a process. This counter indicates a possible handle leak if the number is greater than 10,000.
Process\Thread Count This measures the number of threads currently active in a process. There may be a thread leak if this number is more than 500 between the minimum and maximum number of threads.

Process\Private Bytes This indicates the amount of memory that this process has allocated that cannot be shared with other processes. If the value is greater than 250 between the minimum and maximum number of threads, there may be a memory leak.

Wrapping Up

Now you know what counters the Service Support engineers at Microsoft use to diagnose various bottlenecks. Of course, you will most likely come up with your own set of favorite counters tailored to suit your specific needs. You may want to save time by not having to add all your favorite counters manually each time you need to monitor your servers. Fortunately, there is an option in the Performance Monitor that allows you to save all your counters in a template for later use.

You may still be wondering whether you should run Performance Monitor locally or remotely. And exactly what will the performance hit be when running Performance Monitor locally? This all depends on your specific environment. The performance hit on the server is almost negligible if you set intervals to at least five minutes.

You may want to run Performance Monitor locally if you know there is a performance issue on the server, since Performance Monitor may not be able to capture data from a remote machine when it is running out of resources on the server. Running it remotely from a central machine is really best suited to situations when you want to monitor or baseline multiple servers.