Print this Page **SQL-Server-Performance.com**

Identify SQL Server Hardware Bottlenecks

By : Brad McGehee

Performance Audit Checklist

Counter Name	Average	Minimum	Maximum
Memory: Pages/sec			
Memory: Available Bytes			
Physical Disk: % Disk time			
Physical Disk: Avg. Disk Queue Length			
Processor: % Processor Time			
System: Processor Queue Length			
SQL Server Buffer: Buffer Cache Hit Ratio			
SQL Server General: User Connections			

Enter your results in the table above.

Use Performance Monitor to Help Identify SQL Server Hardware Bottlenecks

The best place to start your SQL Server performance audit is to begin with the Performance Monitor (System Monitor). By monitoring a few key counters over a 24 hour period, you should get a pretty good feel for any major hardware bottlenecks your SQL Server is experiencing.

Ideally, you should use Performance Monitor to create a log of key counters for a period of 24 hours. You will want to select a "typical" 24 hour period when it comes to deciding when to create your Performance Monitor log. For example, pick a typical business day, not a weekend or holiday.

Once you have captured 24 hours of Performance Monitor data in a log, display the recommended counters in the Graph mode of Performance Monitor, and then record the average, minimum, and maximum values in the table above. Once you have done this, then compare your results with the analysis below. By comparing your results with the recommendations below, you should be able to quickly identify any potential hardware bottlenecks your SQL Server is experiencing.

How to Interpret Key Performance Monitor Counters

Below is a discussions of the various key Performance Monitor counters, their recommended values, and some options for helping to identify and resolve the hardware bottlenecks. Note that I have limited the number of Performance Monitor counters to watch. I have done so because our goal in this article is to find the easy and obvious performance problems. Many other Performance Monitor counters can be found discussed elsewhere on this website.

Memory: Pages/sec

This counter measures the number of pages per second that are paged out of RAM to disk, or paged into RAM from disk. The more paging that occurs, the more I/O overhead your server experiences, which in turn can decrease the performance of SQL Server. Your goal is to try to keep paging to a minimum, not to eliminate it.

Assuming that SQL Server is the only major application running on your server, then this figure should ideally average between zero and 20. You will most likely see spikes much greater than 20, which is normal. They key here is keeping the average pages per second less than 20.

If your server is averaging more than 20 pages per second, one of the more likely causes of this is a memory bottleneck due to a lack of needed RAM. Generally speaking, the more RAM a server has, the less paging it has to perform.

In most cases, on a physical server dedicated to SQL Server with an adequate amount of RAM, paging will average less than 20. An adequate amount of RAM for SQL Server is a server that has a Buffer Hit Cache Ratio (described in more detail later) of 99% and higher. If you have a SQL Server that has a Buffer Hit Cache Ratio of 99% or higher for a period of 24 hours, but you are getting an average paging level of over 20 during this same time period, this may indicate that you are running other applications on the physical server other than SQL Server. If this is the case, you should ideally remove those applications, allowing SQL Server to be the only major application on the physical server.

If your SQL Server is not running any other applications, and paging exceeds 20 on average for a 24 hour period, this may mean that you have changed the SQL Server memory settings. SQL Server should be configured so that it is set to the "Dynamically configure SQL Server memory" option, and the "Maximum Memory" setting should be set to the highest level. For optimum performance, SQL Server should be allowed to take as much RAM as it wants for its own use without having to compete for RAM with other applications.

Memory: Available Bytes

Another way to check to see if your SQL Server has enough physical RAM is to check the Memory Object: Available Bytes counter. This value should be greater than 5MB. If not, then your SQL Server needs more physical RAM. On a server dedicated to SQL Server, SQL Server attempts to maintain from 4-10MB of free physical memory. The remaining physical RAM is used by the operating system and SQL Server. When the amount of available bytes is near 5MB, or lower, most likely SQL Server is experiencing a performance hit due to lack of memory. When this happens, you either need to increase the amount of physical RAM in the server, reduce the load on the server, or change your SQL Server's memory configuration settings appropriately.

Physical Disk: % Disk Time

This counter measures how busy a physical array is (not a logical partition or individual disks in an array). It provides a good relative measure of how busy your arrays are.
As a rule of thumb, the % Disk Time counter should run less than 55%. If this counter exceeds 55% for continuous periods (over 10 minutes or so during your 24 hour monitoring period), then your SQL Server may be experiencing an I/O bottleneck. If you see this behavior only occasionally in your 24 hour monitoring period, I wouldn't worry too much, but if it happens often (say, several times an hour), then I would start looking into finding ways to increase the I/O performance on the server, or to reduce the load on the server. Some ways to boost disk I/O include adding drives to an array (if you can), getting faster drives, adding cache memory to the controller card (if you can), using a different version of RAID, or getting a faster controller.

Before using this counter under NT 4.0, be sure to manually turn it on by going to the NT Command Prompt and entering the following: "diskperf -y", and then rebooting your server.

This is required to turn on the disk counters on for the first time under Windows NT 4.0. If you are running Windows 2000, this counter is turned on by default.

Physical Disk: Avg. Disk Queue Length

Besides watching the Physical Disk: % Disk Time counter, you will also want to watch the Avg. Disk Queue Length counter as well. If it exceeds 2 for continuous periods (over 10 minutes or so during your 24 hour monitoring period) for each disk drive in an array, then you may have an I/O bottleneck for that array. Like the Physical Disk: % Disk Time counter, if this happens occasionally in your 24 hour monitoring period, I wouldn't worry too much, but if it happens often, then I would start looking into finding ways to increase the I/O performance on the server, as described previously.

You will need to calculate this figure because Performance Monitor does not know how many physical drives are in your array. For example, if you have an array of 6 physical disks, and the Avg. Disk Queue Length is 10 for a particular array, then the actual Avg. Disk Queue Length for each drive is 1.66 (10/6=1.66), which is well within the recommended 2 per physical disk.

Before using this counter under NT 4.0, be sure to manually turn it on by going to the NT Command Prompt and entering the following: "diskperf -y", and then rebooting your server. This is required to turn on the disk counters on for the first time under Windows NT 4.0. If you are running Windows 2000, this counter is turned on by default.

Use both the % Disk Time and the Avg. Disk Queue Length counters together to help you decide if your server is experiencing an I/O bottleneck. For example, if you see many time periods where the % Disk Time is over 55% and when the Avg. Disk Queue Length counter is over 2 per physical disk, you can be confident the server is having a I/O bottleneck.

Processor: % Processor Time

The Processor Object: % Processor Time counter, is available for each CPU (instance), and measures the utilization of each individual CPU. This same counter is also available for all of the CPUs (total). This is the key counter to watch for CPU utilization. If the % Total Processor Time (total) counter exceeds 80% for continuous periods (over 10 minutes or so during your 24 hour monitoring period), then you may have a CPU bottleneck. If these busy periods are only occur occasionally, and you think you can live with them, that's OK. But if they occur often, you may want to consider reducing the load on the server, getting faster CPUs, getting more CPUs, or getting CPUs that have a larger on-board L2 cache.

System: Processor Queue Length

Along with the Processor: % Processor Time counter, you will also want to monitor the Processor Queue Length counter. If it exceeds 2 per CPU for continuous periods (over 10 minutes or so during your 24 hour monitoring period), then you probably have a CPU bottleneck. For example, if you have 4 CPUs in your server, the Processor Queue Length should not exceed a total of 8 for the entire server.

If the Processor Queue Length regularly exceeds the recommended maximum, but the CPU utilization is not correspondingly as high (which is typical), then consider reducing the SQL Server "max worker threads" configuration setting. It is possible the reason that the Processor Queue Length is high is because there are an excess number of worker threads waiting to take their turn. By reducing the number of "maximum worker threads", what you are doing is forcing thread pooling to kick in (if it hasn't already), or to take greater advantage of thread pooling.

Use both the Processor Queue Length and the % Total Process Time counters together to determine if you have a CPU bottleneck. If both indicators are exceeding their recommended amounts during the same continuous time periods, you can be assured there is a CPU bottleneck.

SQL Server Buffer: Buffer Cache Hit Ratio

This SQL Server Buffer: Buffer Cache Hit Ratio counter indicates how often SQL Server goes to the buffer, not the hard disk, to get data. In OLTP applications, this ratio should exceed 90%, and ideally be over 99%. If your buffer cache hit ratio is lower than 90%, you need to go out and buy more RAM today. If the ratio is between 90% and 99%, then you should seriously consider purchasing more RAM, as the closer you get to 99%, the faster your SQL Server will perform. In some cases, if your database is very large, you may not be able to get close to 99%, even if you put the maximum amount of RAM in your server. All you can do is add as much as you can, and then live with the consequences.

In OLAP applications, the ratio can be much less because of the nature of how OLAP works. In any case, more RAM should increase the performance of SQL Server.

SQL Server General: User Connections

Since the number of users using SQL Server affects its performance, you may want to keep an eye on the SQL Server General Statistics Object: User Connections counter. This shows the number of user connections, not the number of users, that currently are connected to SQL Server.

If this counter exceeds 255, then you may want to boost the SQL Server configuration setting, "Maximum Worker Threads" to a figure higher than the default setting of 255. If the number of connections exceeds the number of available worker threads, then SQL Server will begin to share worker threads, which can hurt performance. The setting for "Maximum Worker Threads" should be higher than the maximum number of user connections your server ever reaches.

Where to Go From Here

While there are a lot more counters than the ones you find on this page, these cover the key counters that you need to monitor during your Performance Audit. Once you have completed your Performance Monitor analysis, use the recommendations presented here, and later in this article series, to make the necessary changes to get your SQL Server performing as it should.