

How to Perform a SQL Server Performance Audit (2003-03-27) - Contributed by Brad McGehee

This lengthy article written by Brad, explains how to check the performance of your SQL Server.

If you have been a DBA for long, then you will know that performance tuning SQL Server is not an exact science. And even if it were, it would still be difficult to find the "optimal" configuration for the "optimal" performance. This is because there are few absolutes when it comes to performance tuning. For example, while one particular performance-tuning tip may help boost performance one way, at the same time it might hurt performance in another way.

Over the last five years that I have been a SQL Server DBA, I have put together a mental checklist that I use when it comes to performance tuning SQL Servers. I use this checklist as a guide when I perform a "performance audit" a (new to me) SQL Server for the first time. I often get requests to "look over" a SQL Server and to offer suggestions on how to boost its performance. Until now, I have never really written down the process I go through. But as I have begun to do more and more performance tuning consulting, I now have decided that it is time to put this performance audit checklist on paper. Hopefully, you will find it as useful to you as I find it is to me.

The SQL Server Performance Audit

The goal of this performance audit checklist is to help you, in a quasi-scientific way, is to help you identify any obvious performance problems with your SQL Server. As I mentioned above, SQL Server performance tuning can be difficult. What I am trying to do with this checklist is to identify all of the "easy" SQL Server performance problems, leaving the hard ones for a later time. I am doing this because is it easy to confuse the easy and the hard SQL Server performance tuning decisions. By creating a list of the "easy" performance tuning areas, it is easier to focus on getting the easy ones out of the way, and once they are out of the way, then you can focus on the more difficult ones.

One of the advantages of using this checklist to perform a SQL Server performance audit is that is will not only tell you what can do to easily boost current performance, it also can be used to help you know what you have already done correctly. In some cases, the choices you have made for your SQL Server may be deliberately different than the recommendations found on this checklist. In other words, you have purposely made a specific decision not to follow common SQL Server performance tuning practices. In some cases, you may be right. Not all SQL Server performance recommendations are applicable to all situations. In other cases, you may have made a decision based on resource limitations, such as not having the money to purchase the necessary hardware to carry your load.

If that's the case, then you have no choice but to live with this. And in other cases, the decisions you have made may be due to political reasons, which may or may not be able to be changed. In any event, you need to do what you can, using this performance audit to identify those areas that you can change, and making those changes to boost your SQL Server's performance.

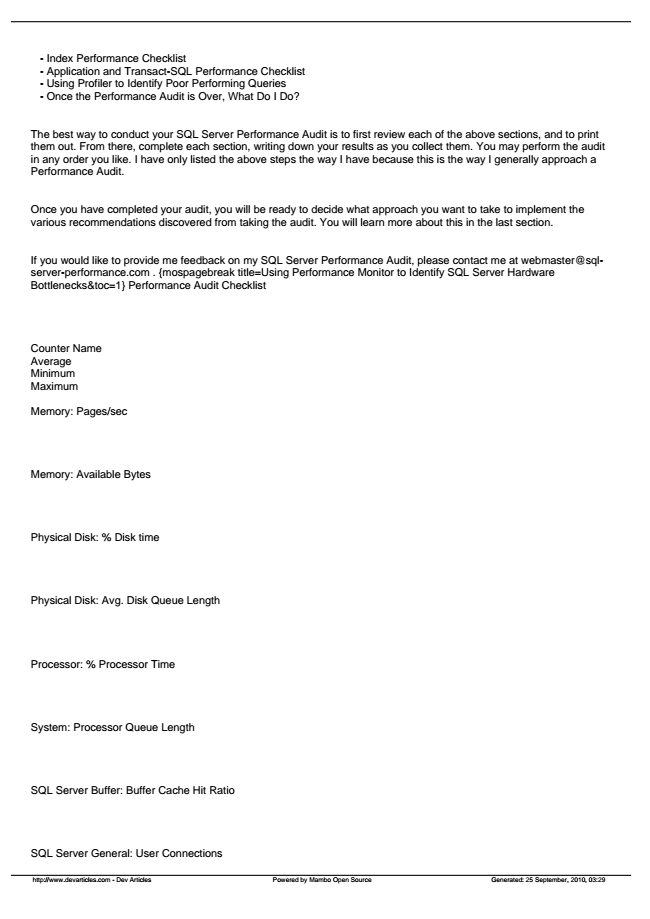
Ideally, you should perform this audit on each of your SQL Servers. If you have many of them, this could take some time. I would suggest that you start on the servers that are currently producing the most performance problems, and the working your way to the rest of the servers as you have time.

Once you complete your performance audit, you still aren’t done. Remember, these are the easy ones. Once you have the easy ones out of the way, then you can begin to devote your time to working on the harder performance issues. And that’s another article for another time.

How to Conduct Your SQL Server Performance Audit

To make your SQL Server Performance Audit easy to perform, I have divided it into several sections. They include:

- Using Performance Monitor to Identify SQL Server Hardware Bottlenecks - Server Hardware Performance Checklist - Operating System Performance Checklist - SQL Server 2000 Configuration Performance Checklist - Database Configuration Settings Performance Checklist http://www.devarticles.com - Dev Articles Powered by Mambo Open Source Generated: 25 September, 2010, 03:29



- Index Performance Checklist - Application and Transact-SQL Performance Checklist - Using Profiler to Identify Poor Performing Queries - Once the Performance Audit is Over, What Do I Do?

The best way to conduct your SQL Server Performance Audit is to first review each of the above sections, and to print them out. From there, complete each section, writing down your results as you collect them. You may perform the audit in any order you like. I have only listed the above steps the way I have because this is the way I generally approach a Performance Audit.

Once you have completed your audit, you will be ready to decide what approach you want to take to implement the various recommendations discovered from taking the audit. You will learn more about this in the last section.

If you would like to provide me feedback on my SQL Server Performance Audit, please contact me at webmaster@sql- server-performance.com . {mospagebreak title=Using Performance Monitor to Identify SQL Server Hardware Bottlenecks&toc=1} Performance Audit Checklist

Counter Name Average Minimum Maximum

Memory: Pages/sec

Memory: Available Bytes

Physical Disk: % Disk time

Physical Disk: Avg. Disk Queue Length

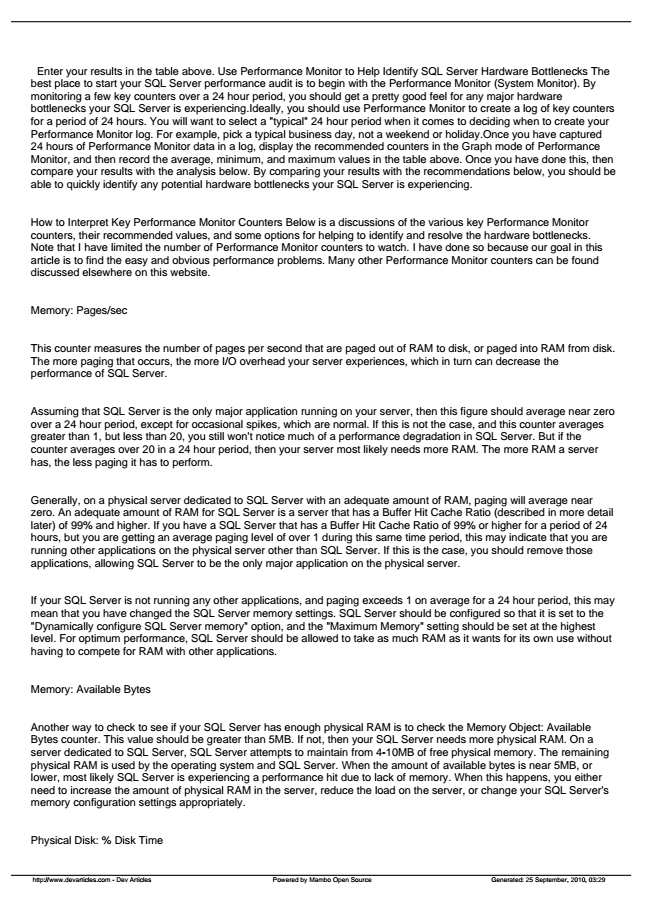
Processor: % Processor Time

System: Processor Queue Length

SQL Server Buffer: Buffer Cache Hit Ratio

SQL Server General: User Connections

http://www.devarticles.com - Dev Articles Powered by Mambo Open Source Generated: 25 September, 2010, 03:29



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.

Assuming that SQL Server is the only major application running on your server, then this figure should average near zero over a 24 hour period, except for occasional spikes, which are normal. If this is not the case, and this counter averages greater than 1, but less than 20, you still won't notice much of a performance degradation in SQL Server. But if the counter averages over 20 in a 24 hour period, then your server most likely needs more RAM. The more RAM a server has, the less paging it has to perform.

Generally, on a physical server dedicated to SQL Server with an adequate amount of RAM, paging will average near zero. 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 1 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 remove those applications, allowing SQL Server to be the only major application on the physical server.

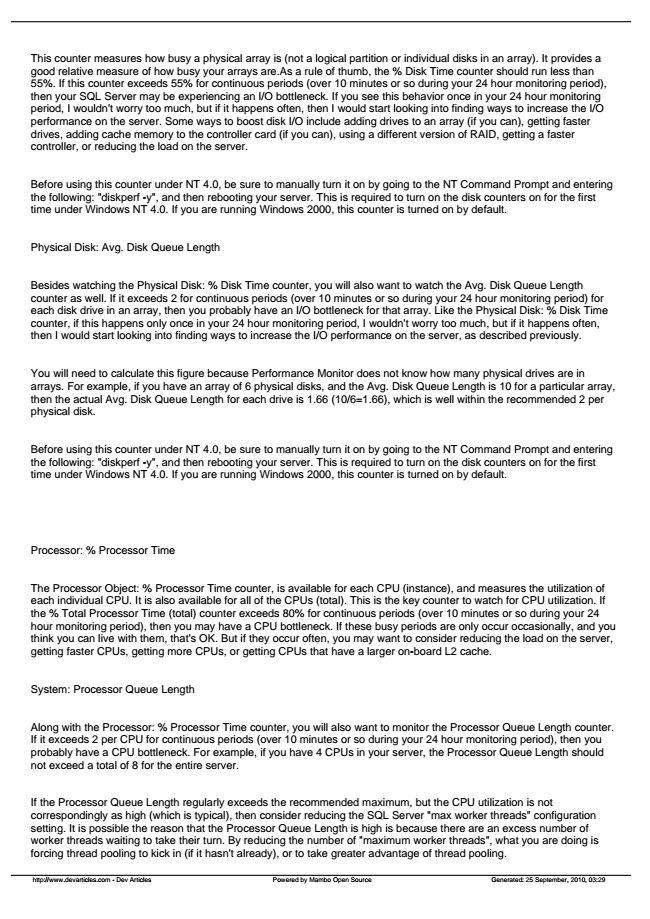
If your SQL Server is not running any other applications, and paging exceeds 1 on average for a 24 hour period, this may mean that you have changed the SQL Server memory settings. SQL Server should be configured so that it is set to the "Dynamically configure SQL Server memory" option, and the "Maximum Memory" setting should be set at 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

http://www.devarticles.com - Dev Articles Powered by Mambo Open Source Generated: 25 September, 2010, 03:29



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 once 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. 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, getting a faster controller, or reducing the load on the server.

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 probably have an I/O bottleneck for that array. Like the Physical Disk: % Disk Time counter, if this happens only once 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 arrays. 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.

Processor: % Processor Time

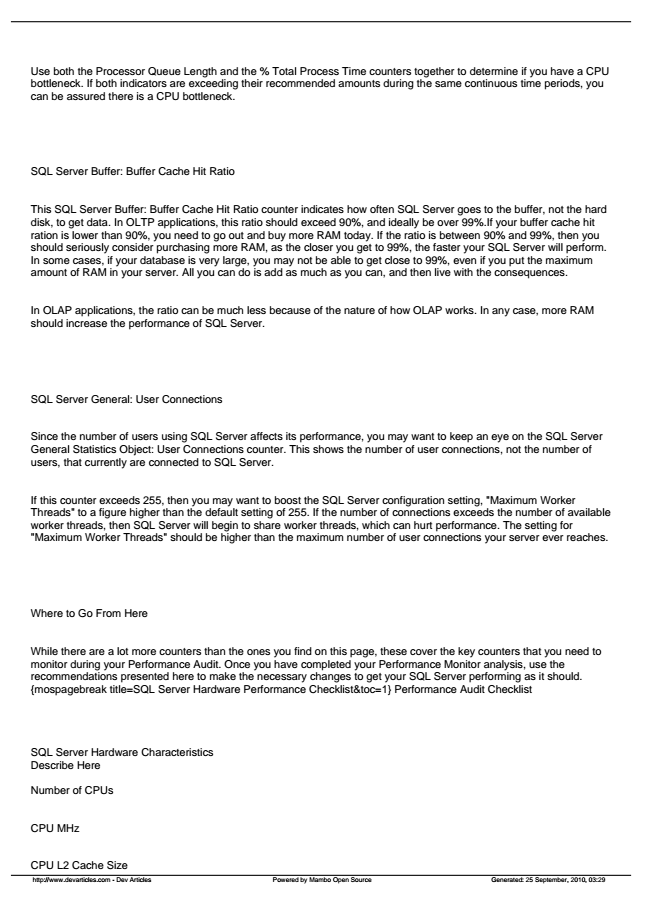
The Processor Object: % Processor Time counter, is available for each CPU (instance), and measures the utilization of each individual CPU. It is also available for all of the CPUs (total). This is the key counter to watch for CPU utilization. If the % Total Processor Time (total) counter exceeds 80% 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.

http://www.devarticles.com - Dev Articles Powered by Mambo Open Source Generated: 25 September, 2010, 03:29



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 ration 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 to make the necessary changes to get your SQL Server performing as it should. {mospagebreak title=SQL Server Hardware Performance Checklist&toc=1} Performance Audit Checklist

SQL Server Hardware Characteristics Describe Here

Number of CPUs

CPU MHz

CPU L2 Cache Size

http://www.devarticles.com - Dev Articles Powered by Mambo Open Source Generated: 25 September, 2010, 03:29



Physical RAM Amount

Total Amount of Available Drive Space on Server

Total Number of Physical Drives in Each Array

RAID Level of Array Used for SQL Server Databases

Hardware vs. Software RAID

Disk Fragmentation Level

Location of Operating System

Location of SQL Server Executables

Location of Swap File

Location of tempdb Database

Location of System Databases

Location of User Databases

Location of Log Files

Number of Disk Controllers in Server

Type of Disk Controllers in Server

Size of Cache in Disk Controllers in Server

Is Write Back Cache in Disk Controller On or Off?

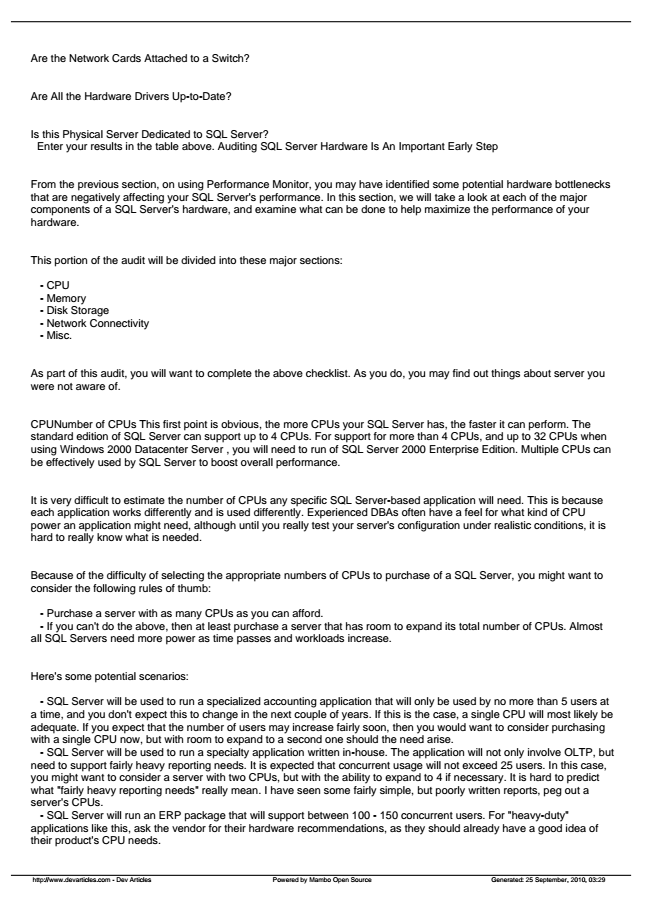
Speed of Disk Drives

How Many Network Cards Are in Server?

What is the Speed of the Network Cards in Server?

Are the Network Cards Hard-Coded for Speed/Duplex?

http://www.devarticles.com - Dev Articles Powered by Mambo Open Source Generated: 25 September, 2010, 03:29



Are the Network Cards Attached to a Switch?

Are All the Hardware Drivers Up-to-Date?

Is this Physical Server Dedicated to SQL Server?

Enter your results in the table above. Auditing SQL Server Hardware Is An Important Early Step

From the previous section, on using Performance Monitor, you may have identified some potential hardware bottlenecks that are negatively affecting your SQL Server's performance. In this section, we will take a look at each of the major components of a SQL Server's hardware, and examine what can be done to help maximize the performance of your hardware.

This portion of the audit will be divided into these major sections:

- CPU - Memory - Disk Storage - Network Connectivity - Misc.

As part of this audit, you will want to complete the above checklist. As you do, you may find out things about server you were not aware of.

CPUNumber of CPUs This first point is obvious, the more CPUs your SQL Server has, the faster it can perform. The standard edition of SQL Server can support up to 4 CPUs. For support for more than 4 CPUs, and up to 32 CPUs when using Windows 2000 Datacenter Server , you will need to run of SQL Server 2000 Enterprise Edition. Multiple CPUs can be effectively used by SQL Server to boost overall performance.

It is very difficult to estimate the number of CPUs any specific SQL Server-based application will need. This is because each application works differently and is used differently. Experienced DBAs often have a feel for what kind of CPU power an application might need, although until you really test your server's configuration under realistic conditions, it is hard to really know what is needed.

Because of the difficulty of selecting the appropriate numbers of CPUs to purchase of a SQL Server, you might want to consider the following rules of thumb:

- Purchase a server with as many CPUs as you can afford. - If you can't do the above, then at least purchase a server that has room to expand its total number of CPUs. Almost all SQL Servers need more power as time passes and workloads increase.

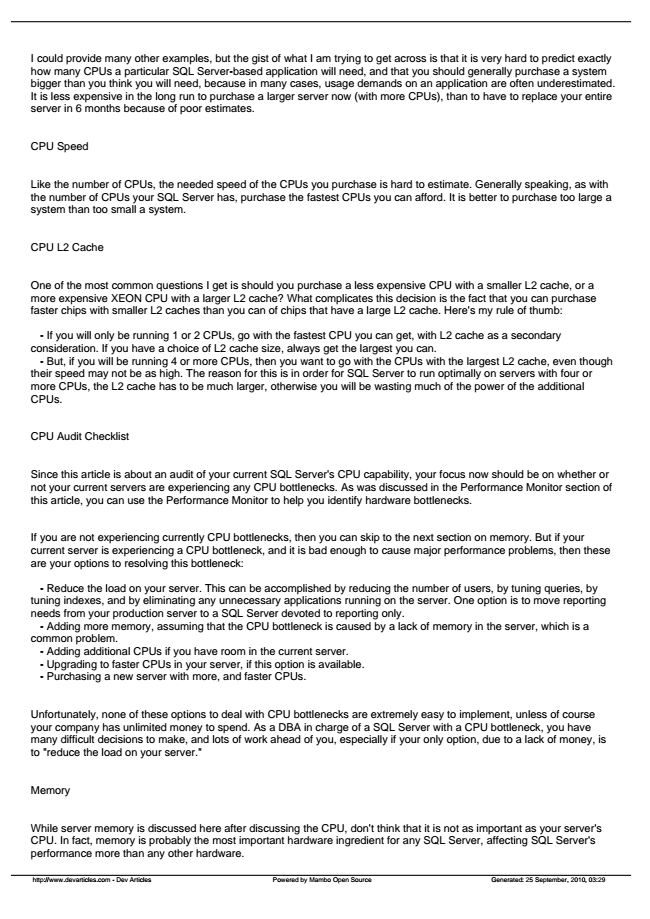
Here's some potential scenarios:

- SQL Server will be used to run a specialized accounting application that will only be used by no more than 5 users at a time, and you don't expect this to change in the next couple of years. If this is the case, a single CPU will most likely be adequate. If you expect that the number of users may increase fairly soon, then you would want to consider purchasing with a single CPU now, but with room to expand to a second one should the need arise.

- SQL Server will be used to run a specialty application written in-house. The application will not only involve OLTP, but need to support fairly heavy reporting needs. It is expected that concurrent usage will not exceed 25 users. In this case, you might want to consider a server with two CPUs, but with the ability to expand to 4 if necessary. It is hard to predict what "fairly heavy reporting needs" really mean. I have seen some fairly simple, but poorly written reports, peg out a server's CPUs.

- SQL Server will run an ERP package that will support between 100 - 150 concurrent users. For "heavy-duty" applications like this, ask the vendor for their hardware recommendations, as they should already have a good idea of their product's CPU needs.

http://www.devarticles.com - Dev Articles Powered by Mambo Open Source Generated: 25 September, 2010, 03:29



I could provide many other examples, but the gist of what I am trying to get across is that it is very hard to predict exactly how many CPUs a particular SQL Server-based application will need, and that you should generally purchase a system bigger than you think you will need, because in many cases, usage demands on an application are often underestimated. It is less expensive in the long run to purchase a larger server now (with more CPUs), than to have to replace your entire server in 6 months because of poor estimates.

CPU Speed

Like the number of CPUs, the needed speed of the CPUs you purchase is hard to estimate. Generally speaking, as with the number of CPUs your SQL Server has, purchase the fastest CPUs you can afford. It is better to purchase too large a system than too small a system.

CPU L2 Cache

One of the most common questions I get is should you purchase a less expensive CPU with a smaller L2 cache, or a more expensive XEON CPU with a larger L2 cache? What complicates this decision is the fact that you can purchase faster chips with smaller L2 caches than you can of chips that have a large L2 cache. Here's my rule of thumb:

- If you will only be running 1 or 2 CPUs, go with the fastest CPU you can get, with L2 cache as a secondary consideration. If you have a choice of L2 cache size, always get the largest you can.

- But, if you will be running 4 or more CPUs, then you want to go with the CPUs with the largest L2 cache, even though their speed may not be as high. The reason for this is in order for SQL Server to run optimally on servers with four or more CPUs, the L2 cache has to be much larger, otherwise you will be wasting much of the power of the additional CPUs.

CPU Audit Checklist

Since this article is about an audit of your current SQL Server's CPU capability, your focus now should be on whether or not your current servers are experiencing any CPU bottlenecks. As was discussed in the Performance Monitor section of this article, you can use the Performance Monitor to help you identify hardware bottlenecks.

If you are not experiencing currently CPU bottlenecks, then you can skip to the next section on memory. But if your current server is experiencing a CPU bottleneck, and it is bad enough to cause major performance problems, then these are your options to resolving this bottleneck:

- Reduce the load on your server. This can be accomplished by reducing the number of users, by tuning queries, by tuning indexes, and by eliminating any unnecessary applications running on the server. One option is to move reporting needs from your production server to a SQL Server devoted to reporting only.

- Adding more memory, assuming that the CPU bottleneck is caused by a lack of memory in the server, which is a common problem.

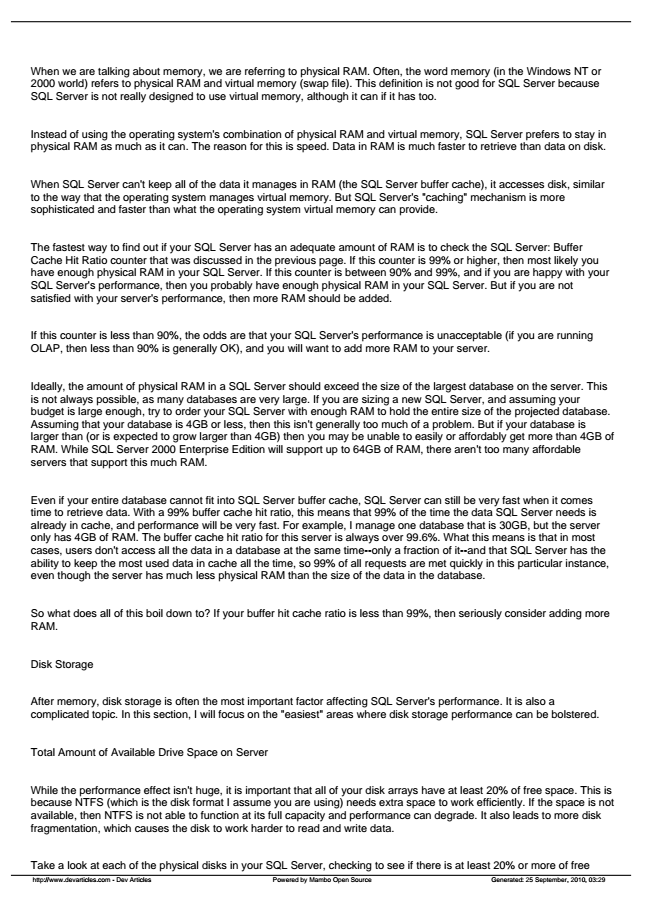
- Adding additional CPUs if you have room in the current server. - Upgrading to faster CPUs in your server, if this option is available. - Purchasing a new server with more, and faster CPUs.

Unfortunately, none of these options to deal with CPU bottlenecks are extremely easy to implement, unless of course your company has unlimited money to spend. As a DBA in charge of a SQL Server with a CPU bottleneck, you have many difficult decisions to make, and lots of work ahead of you, especially if your only option, due to a lack of money, is to "reduce the load on your server."

Memory

While server memory is discussed here after discussing the CPU, don't think that it is not as important as your server's CPU. In fact, memory is probably the most important hardware ingredient for any SQL Server, affecting SQL Server's performance more than any other hardware.

http://www.devarticles.com - Dev Articles Powered by Mambo Open Source Generated: 25 September, 2010, 03:29



When we are talking about memory, we are referring to physical RAM. Often, the word memory (in the Windows NT or 2000 world) refers to physical RAM and virtual memory (swap file). This definition is not good for SQL Server because SQL Server is not really designed to use virtual memory, although it can if it has too.

Instead of using the operating system's combination of physical RAM and virtual memory, SQL Server prefers to stay in physical RAM as much as it can. The reason for this is speed. Data in RAM is much faster to retrieve than data on disk.

When SQL Server can't keep all of the data it manages in RAM (the SQL Server buffer cache), it accesses disk, similar to the way that the operating system manages virtual memory. But SQL Server's "caching" mechanism is more sophisticated and faster than what the operating system virtual memory can provide.

The fastest way to find out if your SQL Server has an adequate amount of RAM is to check the SQL Server: Buffer Cache Hit Ratio counter that was discussed in the previous page. If this counter is 99% or higher, then most likely you have enough physical RAM in your SQL Server. If this counter is between 90% and 99%, and if you are happy with your SQL Server's performance, then you probably have enough physical RAM in your SQL Server. But if you are not satisfied with your server's performance, then more RAM should be added.

If this counter is less than 90%, the odds are that your SQL Server's performance is unacceptable (if you are running OLAP, then less than 90% is generally OK), and you will want to add more RAM to your server.

Ideally, the amount of physical RAM in a SQL Server should exceed the size of the largest database on the server. This is not always possible, as many databases are very large. If you are sizing a new SQL Server, and assuming your budget is large enough, try to order your SQL Server with enough RAM to hold the entire size of the projected database. Assuming that your database is 4GB or less, then this isn't generally too much of a problem. But if your database is larger than (or is expected to grow larger than 4GB) then you may be unable to easily or affordably get more than 4GB of RAM. While SQL Server 2000 Enterprise Edition will support up to 64GB of RAM, there aren't too many affordable servers that support this much RAM.

Even if your entire database cannot fit into SQL Server buffer cache, SQL Server can still be very fast when it comes time to retrieve data. With a 99% buffer cache hit ratio, this means that 99% of the time the data SQL Server needs is already in cache, and performance will be very fast. For example, I manage one database that is 30GB, but the server only has 4GB of RAM. The buffer cache hit ratio for this server is always over 99.6%. What this means is that in most cases, users don't access all the data in a database at the same time--only a fraction of it--and that SQL Server has the ability to keep the most used data in cache all the time, so 99% of all requests are met quickly in this particular instance, even though the server has much less physical RAM than the size of the data in the database.

So what does all of this boil down to? If your buffer hit cache ratio is less than 99%, then seriously consider adding more RAM.

Disk Storage

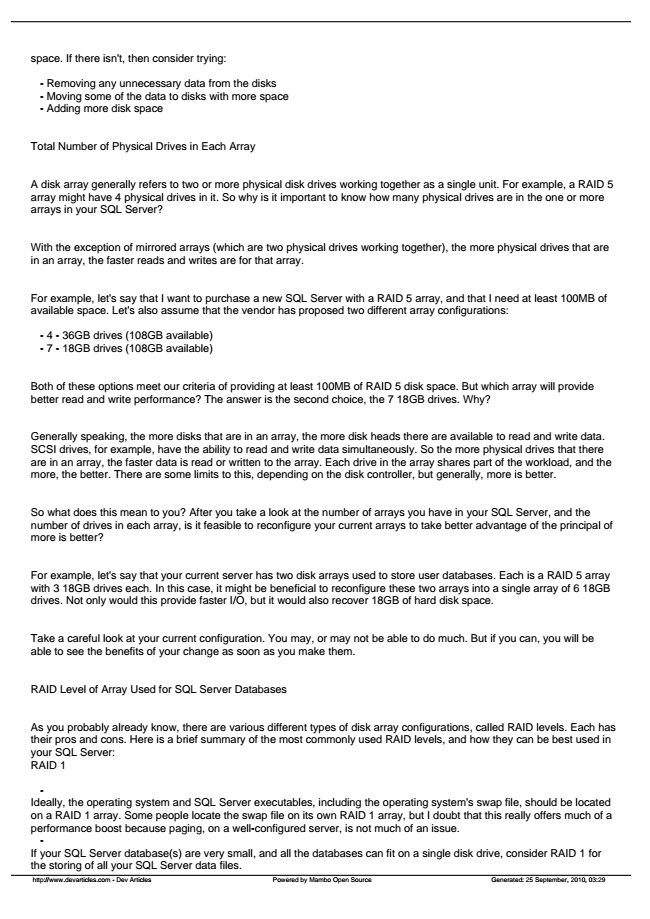
After memory, disk storage is often the most important factor affecting SQL Server's performance. It is also a complicated topic. In this section, I will focus on the "easiest" areas where disk storage performance can be bolstered.

Total Amount of Available Drive Space on Server

While the performance effect isn't huge, it is important that all of your disk arrays have at least 20% of free space. This is because NTFS (which is the disk format I assume you are using) needs extra space to work efficiently. If the space is not available, then NTFS is not able to function at its full capacity and performance can degrade. It also leads to more disk fragmentation, which causes the disk to work harder to read and write data.

Take a look at each of the physical disks in your SQL Server, checking to see if there is at least 20% or more of free

http://www.devarticles.com - Dev Articles Powered by Mambo Open Source Generated: 25 September, 2010, 03:29



space. If there isn't, then consider trying:

- Removing any unnecessary data from the disks - Moving some of the data to disks with more space - Adding more disk space

Total Number of Physical Drives in Each Array

A disk array generally refers to two or more physical disk drives working together as a single unit. For example, a RAID 5 array might have 4 physical drives in it. So why is it important to know how many physical drives are in the one or more arrays in your SQL Server?

With the exception of mirrored arrays (which are two physical drives working together), the more physical drives that are in an array, the faster reads and writes are for that array.

For example, let's say that I want to purchase a new SQL Server with a RAID 5 array, and that I need at least 100MB of available space. Let's also assume that the vendor has proposed two different array configurations:

- 4 - 36GB drives (108GB available) - 7 - 18GB drives (108GB available)

Both of these options meet our criteria of providing at least 100MB of RAID 5 disk space. But which array will provide better read and write performance? The answer is the second choice, the 7 18GB drives. Why?

Generally speaking, the more disks that are in an array, the more disk heads there are available to read and write data. SCSI drives, for example, have the ability to read and write data simultaneously. So the more physical drives that there are in an array, the faster data is read or written to the array. Each drive in the array shares part of the workload, and the more, the better. There are some limits to this, depending on the disk controller, but generally, more is better.

So what does this mean to you? After you take a look at the number of arrays you have in your SQL Server, and the number of drives in each array, is it feasible to reconfigure your current arrays to take better advantage of the principal of more is better?

For example, let's say that your current server has two disk arrays used to store user databases. Each is a RAID 5 array with 3 18GB drives each. In this case, it might be beneficial to reconfigure these two arrays into a single array of 6 18GB drives. Not only would this provide faster I/O, but it would also recover 18GB of hard disk space.

Take a careful look at your current configuration. You may, or may not be able to do much. But if you can, you will be able to see the benefits of your change as soon as you make them.

RAID Level of Array Used for SQL Server Databases

As you probably already know, there are various different types of disk array configurations, called RAID levels. Each has their pros and cons. Here is a brief summary of the most commonly used RAID levels, and how they can be best used in your SQL Server: RAID 1

- Ideally, the operating system and SQL Server executables, including the operating system's swap file, should be located on a RAID 1 array. Some people locate the swap file on its own RAID 1 array, but I doubt that this really offers much of a performance boost because paging, on a well-configured server, is not much of an issue.

- If your SQL Server database(s) are very small, and all the databases can fit on a single disk drive, consider RAID 1 for the storing of all your SQL Server data files.

http://www.devarticles.com - Dev Articles Powered by Mambo Open Source Generated: 25 September, 2010, 03:29