SQL Server Hardware Tuning and Performance Monitoring

(2003-08-14) - Contributed by Sayed Geneidy

In this article, Sayed will lead the way in presenting you guys with a performance enhancing solution for your system. Discover how to optimise your system by investigating what hardware you really need for your applications.

The hardware configuration that SQL Server runs on top of can make or break you. How do you know how much hardware is really needed by your applications? How do you know if your applications have grown to the point where they are now overloading your system?

In this article, I will give you the information you need to solve the specific bottleneck or to resolve future bottlenecking issues. To find out your hardware bottleneck is to use Performance Monitor. Using Performance Monitor, we can identify issues in each of a set of performance objects.

In the following lines we will see what the best way for hardware tuning and performance monitoring is:

I would like to say that the subject of SQL server performance tuning is big and complex. So, I will try to explain today SQL server hardware tuning and performance monitoring.

If you need more covering you can visit "The God Father", of course I mean SQL-server-performance website maintained by Brad. Because it has a lot of valuable information about SQL Server performance tuning subject. At the end of this article I will give you resources to help you and to understand more about this important topic. {mospagebreak title=Performance Monitoring Issues&toc=1} Memory Performance Monitoring

Now that you know the memory limitations in SQL Server imposed by the operating system, what can you do to optimise that memory and determine if you need more? This is where we get into what is popularly called The Art of Performance Monitoring .

Due to the dynamic nature of the memory in SQL Server 7.0 and SQL server 2000, Microsoft removed support for one of the most useful tools available in SQL Server 6.5, DBCC MEMUSAGE. Now, to get this information, you have to monitor a variety of performance counters in the Buffer Manager object and the Cache Manager object since the information is no longer statically mapped in a table like sysconfigures.

While this handy tool is listed as Unsupported now and no longer returns the breakdown of memory usage, try running it. You will see that this unsupported snap shot continues to return the top 20 list of buffered tables and indexes. This can be a very handy list. When analysing performance of an individual application un a test environment, it can be invaluable.

The Art of Performance Monitoring

Performance monitoring is an art, in that it is a combination of talent, experience, knowledge, and sometimes just plain luck. How do you know if you can do it? You have to try, try, and try again. Keep at it, read up on it. Keep a performance monitor continually open against your production server. Here are some good guidelines to get you started.

1. Make sure that you are running your typical processes (SQL Server) and work loads (queries and stored procedures) during your monitoring.

2. Do not just do real-time monitoring of your servers. Capture long running logs. In Windows NT, install the Datalog/Monitor service from the NT Resource Kit; this functionality is available out of the box in Windows 2000.

3. Always have disk counters turned on by running from a command prompt the command DISKPERF Y and then rebooting. Even in a production environment, the overhead is minimal; the last thing you want to do in the middle of a crisis where logical and physical disk counters are necessary is to have to reboot.

4. For routine, daily, desktop monitoring, set up the chart windows with an interval of 18 seconds. In both the Windows NT PerMon and the Windows 2000 MMC SysMon, this will give your chart a Window of 30 minutes. For me, this has proven to be the ideal interval for both seeing the past and minimizing the impact on the server.

5. Utilize SQL Profiler for individual queries and processes in coordination with PerMon or SysMon to get a good picture of the impact of individual queries.

6. Know the terminology of Performance Monitoring. Objects are lists of individual statistics available. An example is the Processor object. A counter is a single statistic that falls under the heading of an object. An example is the per centprocessor Time counter under the Processor object. An instance is further breakdown of a counter statistic into duplicate components. Not all counters will have separate instances. The per centprocessor Time counter has instances for each processor and a _Total instance as a summery of all processor activity.

7. Know your tools. While you may know how to set up a chart in PerMon, learn how to set up a log with Datalog or Performance Log. Other tools to be familiar with are DBCC MEMUSAGE, Task Manager, and SQL Enterprise Manager Current Activity.

8. Do not be afraid to experiment. The BackOffice Resource Kit has tools in it for creating test data (DataSim), creating test databases (DBGen), and simulating loading from multiple clients (SqlLS).

SQL Performance Monitoring and Bottlenecks

Bottlenecks occur when the hardware resources cannot keep up with the demands of the software. For example, when a software process or combination of processes, wants more I/O from a disk than the disk can physically deliver, a bottleneck occurs at the disk. When the CPU subsystem becomes too saturated and processes are waiting, a bottleneck has occurred. Bottlenecks are usually fixed in one of two ways.

The first is to identify the limiting hardware and increase its capabilities. In other words, get a faster hard driver or increase the speed of the CPU. The second way is to make the software processes utilize the hardware more efficiently. This could be done by putting an index on a table so that either the disk I/O necessary to service a query is reduced, or the CPU units necessary to process a join are lessened.

The following are five key areas to monitor when tracking server performance and identifying bottlenecks. Each bottleneck candidate will have varied performance monitoring objects and couners to consider.

- Memory usage SQL Server needs, relative to itself and to the operating system memory. If SQL Server has enough memory but the operating system is starved of memory such that it has to frequently swap through the pagefile to disk, overall performance will suffer dramatically.
- CPU processor utilization High CPU utilization rates indicate the CPU subsystem is underpowered. Solutions could be upgrading the CPU or increasing the number of processors.
- Disk I/O performance Failure of the disk or disk controller to satisfy read or write requirements in a timely manner impacts performance.
- User connections. Improperly setting the number of user connections could rob memory otherwise available to SQL Server.
- Blocking Locks One process keeps another process from accessing or updating data. This is particularly noticeable to users and is the cause of some of your most server performance problems from a user perspective. {mospagebreak title=Memory Tuning Issues&toc=1} Memory Tuning: The Operating System and SQL Server

Start your intensive analysis of memory by looking at two counters,

- Memory: Available Bytes
- Memory: Pages Faults/sec

The Available Bytes counter tells how much memory is available for use by processes.

The Pages Faults/sec counter tells us the number of hard page faults, pages which had to be retrieved from the hard disk since they were not in working memory. It also includes the number of pages written to the hard disk to free space in the working set to support a hard page fault.

A low number for Available Bytes indicates that there may not be enough memory available or processes, including SQL Server, may not be releasing memory. A high number of Pages Faults/sec indicate excessive paging. Further looks at individual instances of process:Page Faults/sec to see if the SQL Server process, for example, has excessive paging. A low rate of Pages Faults/sec (commonly 5-10 per second) is normal, as the operating system will continue to do some house keeping on the working set.

As previously noted, starting with SQL Server 7, memory is auto-tuning by default. In general though, you want to give SQL as much dedicated memory as possible. This is mostly dependent on what other application may be running on the server. By using the sp_configure stored procedure, you can set the values to MIN SERVER MEMORY and MAX SERVER MEMORY to dedicated values.

If SQL Server is the only application on the server, MIN SERVER MEMORY and MAX SERVER MEMORY to the same value. If SQL Server co-exists with one or more applications, lower the MIN SERVER MEMORY setting to count for the memory demands of the other application(s). If the other application fails to start in a timely manner, it may be because SQL Server has been operating at or near the MAX SERVER MEMORY setting and is slow in releasing memory to the new, and now starved, application. In this instance, lower the value of MAX SERVER MEMORY. Obviously, MAX SERVER MEMORY always needs to be greater than or equal to MIN SERVER MEMORY.

NOTE: If you have installed and are running the Full-Text Search support (Microsoft Search service, also known as MSSearch), then you must set the max server memory option manually to leave enough memory for the MSSearch service to run. Microsoft supplies a handy formula here:

Total Virtual Memory (SQL Server MAX + Virtual Memory for Other Processes) = 1.5 * Server Physical Memory.

Once we have tuned the SQL Server memory settings, it is a good idea to decide if you want SQL Server 7/2000 to tune the process memory automatically or have values set for the configuration. For better performance, you can lock the amount of working set memory that SQL Server reserves. The trade-off here is that you may receive out of memory messages from other applications on the same server.

If you do decide to fix the amount of working set memory, two configuration settings are necessary. First, equalize the MIN SERVER MEMORY and MAX SERVER MEMORY settings. Then turn on the SET WORKING SET SIZE configuration flag using sp_configure. MAX SERVER MEMORY should generally not exceed the RAM available for the server.

SQL Server Process Memory Tuning

Once you have gotten the overall OS and SQL server memory tuned, look further at the SQL Server memory usage. Four counters are desirable here:

- process: Working Set:sqlserver
- SQL Server: Buffer Manager: Buffer Cach Hit Ratio
- SQL Server: Buffer Manager: Free Buffers
- SQL Server: Memory Manager: Total Server Memory (KB)

The process: Working Set:sqlserver instance shows the amount of memory that SQL Server is using. If the number is consistently lower than the amount SQL Server is configured to muse by the MIN SERVER MEMORY and MAX SERVER MEMORY options, then SQL Server is configured for too much memory. Otherwise, you may need to increase RAM and MAX SERVER MEMORY.

Buffer Cach Hit Ratio should be consistently greater than 90. This indicates that the data cach supplied 90 per cent of the requests for data. If this value is consistently low, it is a very good indicator that more memory is needed by SQL Server. If Available Bytes is low, this means that we need to add more RAM.

When Free Buffers is low, this means that there is not enough RAM to maintain a consistent amount of data cach. It too, is indicative of a need for more memory.

If Total Server Memory for SQL Server is consistently higher than the overall server memory, it indicates that there is not enough RAM. {mospagebreak title=More Performance Monitoring Issues&toc=1} CPU Performance Monitoring

In CPU performance monitoring, we are going to be using several counters:

- Processor: Per cent Processor Time
- Processor: Per cent Privileged Time
-
Processor: Per cent User Time
- System: Per cent Total Processor Time

Generally, CPU performance monitoring is straightforward. You need to start by monitoring Processor: per cent Processor Time. If you have more than one processor, you should monitor each instance of this counter and also monitor System: per cent Total Processor Time to determine the average for all processors.

Utilization rates consistently above 80-90 per cent may indicate a poorly tuned or designed application. On the other hand, if you have put all the other recommendations of this book into use, they may indicate a need for a more powerful CPU subsystem. In general, I would spend a little bit of time analysing the applications before immediately going out and buying three more processors.

Spending this time experimenting to discover CPU performance problems and correcting them through software improvements will often keep you from just spending money on a more powerful CPU that only covers up poorly written software for little or no time.

If you do see high CPU utilization, you will then want to monitor Processor: Per cent Privileged Time. This is the time spent performing kernel level operations, such as disk I/O. If his counter is consistently above 80-90 per cent and corresponds to high disk performance counters, you may have a disk bottleneck rather than a CPU bottleneck.

What about SQL Server? Processor: Per cent User Time measures the amount of processor time consumed by non-kernel level applications. SQL is such an application. If this is high and you have multiple processes running on a server, you may want to delve further by looking at specific process instances through the instances of the counter Process: Per cent User Time. This can be very useful for occasions such as when our operating system engineers installed new anti-virus software on all our servers. It temporarily brought them to their knees until we were able to determine the culprit through analysing Process:Per cent User Time for the anti-virus software instance.

Disk Tuning and Performance Monitoring

Begin disk performance monitoring by looking at the following counters:

- PhysicalDisk: Per cent Disk Time
- PhysicalDisk: Current Disk Queue Length
- PhysicalDisk: Avg. Disk Queue Length

Applications and systems that are I/O-bound may keep the disk constantly active. This is called disk thrashing.

You should always know how many channels, what types of arrays, how many disks are in each array, and which array/channel your data and transaction logs are located on before you start thinking about disk performance tuning.

The PhysicalDisk: Per cent Disk Time counter monitors the percentage of time that the disk is conducting check the PhysicalDisk: Current Disk Queue Length counter to see the number of requests that are queued up waiting for disk access.

It is important at this point to be familiar with your disk subsystem. If the number of waiting I/O requests is a sustained value more than 1.5 to 2 times the number of spindles making up the physical disk, you have a disk bottleneck. For example, a RAID 5 configuration with seven spindles/disks would be a candidate for disk performance tuning should the Current Disk Queue Length continually rest above 12-14.

To improve performance in this situation, consider adding faster disk drives, moving some processes to an additional controller-disk subsystem, or adding additional disks to a RAID 5 array.

Most disks have one spindle, although RAID devices usually have more. A hardware RAID 5 device appears as one physical disk in Windows NT PerMon or Windows 2000 sysMon. RAID devices created through software appear as multiple instances.

WARNING: The Per cent Disk Time counter can indicate a value greater than 100 per cent if you are using a hardware based RAID configuration. If it does, use the PhysicalDisk: Avg. Disk Queue Length counter to determine the average number of system requests waiting for disk access. Again, this is indicative of a performance problem if a sustained value of 1.5 to 2 times the number of spindles in the array is observed. {mospagebreak title=SQL Server Performance Tuning Resources, Websites and Articles&toc=1} sql-server-performance.com: A website dedicated to SQL Server Performance Tuning. Contains a lot of information to cover performance tuning. A very valuable site :)

Communities & Newsgroups

http://www.microsoft.com/technet/itcommunity/chats/trans/sql/sql0513.asp Review expert answers to user questions with this chat regarding getting the most from SQL Server.

http://www.microsoft.com/technet/itcommunity/chats/trans/sql/pass1124.asp this chat was hosted by Neil MVP/MCSE, Protech Computing Ltd., as well as several MVPs and members of the Microsoft SQL Server team.

http://support.microsoft.com/default.aspx?scid=kb;en-us;322883 The performance tuning guide that this article references provides principles and techniques to help database administrators configure SQL Server 7.0 for maximum performance and to help them determine the cause of poor performance in a SQL Server.

http://support.microsoft.com/default.aspx?scid=kb;en-us;298475 To troubleshoot performance issues, you must complete a series of steps to isolate and determine the cause of the problem. Possible causes include: Blocking System resource contention Application design problems Queries or stored procedures that have...

http://support.microsoft.com/default.aspx?scid=kb;en-us;319942 This article describes the following configuration settings and considerations for their use: Affinity Mask Lightweight Pooling Max Async IO Max Worker Threads Memory Priority Boost Set Working Set Size SQL Server can obtain a very high level of...

Support WebCast

http://support.microsoft.com/default.aspx?scid=kb;en-us;324885 Session Summary During this session, we will show a practical approach to analyze performance tuning problems that involve Microsoft SQL Server lock monitoring data. We will briefly discuss sp_who2, sysprocesses, and syslocks. You should have a firm...

SQL Server 2000 Performance Tuning Books

- Microsoft SQL Server 2000 Performance Tuning Technical Reference
- The SQL Server 2000 Performance Optimization and Tuning Handbook