The information below is to be used as a guide. Just because the counters collected in your environment may be drastically different from those listed below does not mean that there is a problem. I am sure many of these could be debated but I find the information helpful and you can be the judge if it is useful in your environment(s). I am not the author of the content below so if you see something that you disagree with… don't blame me:)  
  
Memory Bottleneck Analysis  
  
Object: - Memory  
Counter: - Available Mbytes  
Preferred Value: - > 20MB  
Description: -  
Reference: - KB 889654  
  
Object: - Memory  
Counter: - Free System Page Table Entries  
Preferred Value: - > 7000  
Description: - Free System Page Table Entries is the number of page table entries not currently in use by the system. If < 7000, consider removing /3GB.  
Reference: - KB 311901  
  
Object: - Memory  
Counter: - Pages/Sec  
Preferred Value: - < 50  
Description: - Pages/sec is the rate at which pages are read from or written to disk to resolve hard page faults. This counter is a primary indicator of the kinds of faults that cause system-wide delays.  
Reference: - Monitoring and Tuning Your Server   
  
Object: - Memory  
Counter: - Pages Input/Sec  
Preferred Value: - < 10  
Description: - Pages Input/sec is the rate at which pages are read from disk to resolve hard page faults.  
Reference: - KB 889654  
  
Object: - Paging File  
Counter: - %Usage  
Preferred Value: - < 70%  
Description: - The amount of the Page File instance in use in percent.   
Reference: - KB 889654  
  
Object: - Paging File  
Counter: - %Usage  
Preferred Value: - < 70%  
Description: - The peak usage of the Page File instance in percent.   
Reference: - KB 889654  
  
Object: - SQL Server:Buffer Manager  
Counter: - Page Life Expectancy  
Preferred Value: - > 300  
Description: - This performance monitor counter tells you, on average, how long data pages are staying in the buffer. If this value gets below 300 seconds, this is a potential indication that your SQL Server could use more memory in order to boost performance.  
Reference: -   
  
Object: - SQL Server:Buffer Manager  
Counter: - Lazy Writes/Sec  
Preferred Value: - < 20  
Description: - This counter tracks how many times a second that the Lazy Writer process is moving dirty pages from the buffer to disk in order to free up buffer space. Generally speaking, this should not be a high value, say more than 20 per second or so. Ideally, it should be close to zero. If it is zero, this indicates that your SQL Server's buffer cache is plenty big and SQL Server doesn't have to free up dirty pages, instead waiting for this to occur during regular checkpoints. If this value is high, then a need for more memory is indicated.  
Reference: -   
  
Object: - SQL Server:Buffer Manager  
Counter: - Checkpoint Pages/Sec  
Preferred Value: - This value is relative, it varies from server to server, we need to compare the average to a base line capture to tell if the value is high or low.  
Description: - When a checkpoint occurs, all dirty pages are written to disk. This is a normal procedure and will cause this counter to rise during the checkpoint process. What you don't want to see is a high value for this counter over time. This can indicate that the checkpoint process is running more often than it should, which can use up valuable server resources. If this has a high figure (and this will vary from server to server), consider adding more RAM to reduce how often the checkpoint occurs, or consider increasing the "recovery interval" SQL Server configuration setting.  
Reference: -   
  
Object: - SQL Server:Buffer Manager  
Counter: - Page reads/sec  
Preferred Value: - < 90  
Description: - Number of physical database page reads issued. 80 – 90 per second is normal, anything that is above indicates indexing or memory constraint.  
Reference: -   
  
Object: - SQL Server:Buffer Manager  
Counter: - Page writes/sec  
Preferred Value: - < 90  
Description: - Number of physical database page writes issued. 80 – 90 per second is normal, anything more we need to check the lazy writer/sec and checkpoint counters, if these counters are also relatively high then, it’s memory constraint.  
Reference: -   
  
Object: - SQL Server:Buffer Manager  
Counter: - Free pages  
Preferred Value: - > 640  
Description: - Total number of pages on all free lists.  
Reference: -   
  
Object: - SQL Server:Buffer Manager  
Counter: - Stolen pages  
Preferred Value: - Varies. Compare with baseline  
Description: - Number of pages used for miscellaneous server purposes (including procedure cache).  
Reference: -   
  
Object: - SQL Server:Buffer Manager  
Counter: - Buffer Cache hit ratio  
Preferred Value: - > 90%  
Description: - Percentage of pages that were found in the buffer pool without having to incur a read from disk.  
Reference: -   
  
Object: - SQL Server:Buffer Manager  
Counter: - Target Server Memory(KB)  
Preferred Value: -   
Description: - Total amount of dynamic memory the server can consume.  
Reference: -   
  
Object: - SQL Server:Buffer Manager  
Counter: - Total Server Memory(KB)  
Preferred Value: -   
Description: - Total amount of dynamic memory (in kilobytes) that the server is using currently  
Reference: -   
  
Disk Bottleneck Analysis  
Object: - PhysicalDisk  
Counter: - Avg. Disk Sec/Read  
Preferred Value: - < 8ms  
Description: - Measure of disk latgency. Avg. Disk sec/Read is the average time, in seconds, of a read of data from the disk.  
More Info:  
Reads or non cached Writes  
Excellent < 08 Msec ( .008 seconds )  
Good < 12 Msec ( .012 seconds )  
Fair < 20 Msec ( .020 seconds )  
Poor > 20 Msec ( .020 seconds )  
Cached Writes Only  
Excellent < 01 Msec ( .001 seconds )  
Good < 02 Msec ( .002 seconds )  
Fair < 04 Msec ( .004 seconds )  
Poor > 04 Msec ( .004 seconds  
Reference: -   
  
Object: - PhysicalDisk  
Counter: - Avg. Disk sec/Write  
Preferred Value: - < 8ms (non cached) < 1ms (cached)  
Description: - Measure of disk latency. Avg. Disk sec/Write is the average time, in seconds, of a write of data to the disk.  
Reference: -   
  
Object: - PhysicalDisk  
Counter: - Avg. Disk Read Queue Length  
Preferred Value: - < 2 \* spindles  
Description: - Avg. Disk Read Queue Length is the average number of read requests that were queued for the selected disk during the sample interval.  
More Info:  
< (2+ no of spindles) Excellent  
< (2\*no of spindles) Good  
< (3\* no of spindles) Fair  
  
Reference - Whitepaper “Performance Monitoring in Windows 2003: Best Practices” by Ben W. Christenbury  
  
Note: If the disk has say 20 disk and it is RAID 10 then no. of spindles = 20/2 = 10. If it is RAID 5 then the no. of spindles = no of disks = 20.  
Reference: -   
  
Object: - PhysicalDisk  
Counter: - Avg. Disk Write Queue Length  
Preferred Value: - < 2 \* spindles  
Description: - Avg. Disk Write Queue Length is the average number of write requests that were queued for the selected disk during the sample interval.  
Reference: -   
  
Object: - SQL Server:Buffer Manager  
Counter: - Page reads/sec  
Preferred Value: - < 90  
Description: - Number of physical database page reads issued. 80 – 90 per second is normal, anything that is above indicates indexing or memory constraint.  
Reference: -   
  
Object: - SQL Server:Buffer Manager  
Counter: - Page writes/sec  
Preferred Value: - < 90  
Description: - Number of physical database page writes issued. 80 – 90 per second is normal, anything more we need to check the lazy writer/sec and checkpoint counters, if these counters are also relatively high then, it’s memory constraint.  
Reference: -   
  
Object: - SQL Server:Buffer Manager  
Counter: - Free pages  
Preferred Value: - > 640  
Description: - Total number of pages on all free lists.  
Reference: -   
  
Object: - SQL Server:Buffer Manager  
Counter: - Stolen pages  
Preferred Value: - Varies. Compare with baseline  
Description: - Number of pages used for miscellaneous server purposes (including procedure cache).  
Reference: -   
  
Object: - SQL Server:Buffer Manager  
Counter: - Buffer Cache hit ratio  
Preferred Value: - > 90%  
Description: - Percentage of pages that were found in the buffer pool without having to incur a read from disk.  
Reference: -   
  
Processor Bottleneck Analysis  
  
Object: - Processor  
Counter: - %Processor Time  
Preferred Value: - < 80%  
Description: - % Processor Time is the percentage of elapsed time that the processor spends to execute a non-Idle thread.  
Reference: -   
  
Object: - Processor  
Counter: - %Privileged Time  
Preferred Value: - < 30% of Total %Processor Time  
Description: - % Privileged Time is the percentage of elapsed time that the process threads spent executing code in privileged mode.   
Reference: -   
  
Object: - Process (sqlservr)  
Counter: - %Processor Time  
Preferred Value: - < 80%  
Description: -  
Reference: -   
  
Object: - Process (sqlservr)  
Counter: - %Privileged Time  
Preferred Value: - < 30% of %Processor Time (sqlservr)  
Description: - Note: Divide the value by number of processors  
Reference: -   
  
Object: - System  
Counter: - Context Switches/sec  
Preferred Value: - < 3000  
Description: - 1500 – 3000 per processor Excellent – Fair  
> 6000 per processor Poor  
Upper limit is about 40,000 at 90 % CPU per CPU  
NOTE: Remember to divide by number of processors  
Reference: -   
  
Object: - System  
Counter: - Processor Queue Length  
Preferred Value: - < 4 per CPU  
Description: - For standard servers with long Quantums  
<= 4 per CPU Excellent  
< 8 per CPU Good  
< 12 per CPU Fair  
Reference: -   
  
Object: - SQLServer:Access Methods  
Counter: - Full Scans / sec  
Preferred Value: - < 1  
Description: - If we see high CPU then we need to invistigate this counter, otherwise if the full scan are on small tables we can ignore this counter. Values greater than 1 or 2 indicates that we are having table / Index page scans. We need to analyze how this can be avoided.  
Reference: -   
  
Object: - SQLServer:Access Methods  
Counter: - Worktables Created/Sec  
Preferred Value: - < 20  
Description: - Number of worktables created in tempdb per second. Worktables are used for queries that use various spools (table spool, index spool, etc).  
Reference: -   
  
Object: - SQLServer:Access Methods  
Counter: - Workfiles Created/Sec  
Preferred Value: - < 20  
Description: - Number of work files created per second. Tempdb workfiles are used in processing hash operations when the amount of data being processed is too big to fit into the available memory. They may be able to reduce this number by making the queries more efficient by adding/changing indexes, adding additional memory, etc.  
Reference: -   
  
Object: - SQLServer:Access Methods  
Counter: - Page Splits/sec  
Preferred Value: - < 20   
Description: - Interesting counter that can lead us to our table / index design. This value needs to be low as possible. If you find out that the number of page splits is high, consider increasing the fillfactor of your indexes. An increased fillfactor helps to reduce page splits because there is more room in data pages before it fills up and a page split has to occur.  
Reference: -   
  
Overall SQL Server Bottleneck Analysis  
  
Object: - SQLServer:General Statistics  
Counter: - User Connections  
Preferred Value: -   
Description: - The number of users currently connected to the SQL Server.  
Reference: -   
  
Object: - SQLServer:General Statistics  
Counter: - Logins/sec  
Preferred Value: - < 2   
Description: - > 2 per second indicates that the application is not correctly using connection pooling.  
  
Reference: -   
  
Object: - SQLServer:General Statistics  
Counter: - Logouts/sec  
Preferred Value: - < 2  
Description: - > 2 per second indicates that the application is not correctly using connection pooling.  
Reference: -   
  
Object: - SQLServer:SQL Statistics  
Counter: - Batch Requests/Sec  
Preferred Value: - < 1000  
Description: - Over 1000 batch requests per second indicate a very busy SQL Server.  
Reference: -   
  
Object: - SQLServer:SQL Statistics  
Counter: - SQL Compilations/sec  
Preferred Value: - < 10% of the number of Batch Requests / sec  
Description: - The number of times per second that SQL Server compilations have occurred. This value needs to be as low as possible. If you see a high value such as over 100, then it’s an indication that there are lots or adhoc queries that are running, might cause CPU  
Reference: -   
  
Object: - SQLServer:SQL Statistics  
Counter: - SQL Re-Compilations/sec  
Preferred Value: - < 10% of the number of SQL Compilations/sec  
Description: - This needs to be nil in our system as much as possible. A recompile can cause deadlocks and compile locks that are not compatible with any locking type.  
Reference: -   
  
Object: - SQL Server:Latches  
Counter: - Average Latch Wait Time (ms)  
Preferred Value: - < 300  
Description: - Average latch wait time (milliseconds) for latch requests that had to wait.  
Reference: -   
  
Transaction Management  
  
Object: - SQL Server:Locks   
Counter: - Number of Deadlocks/sec  
Preferred Value: - < 1  
Description: - The number of lock requests that resulted in a deadlock.  
Reference: -   
  
Object: - SQL Server:Locks   
Counter: - Lock Requests/sec  
Preferred Value: - < 1000  
Description: - Number of requests for a type of lock per second. Lock requests/sec > 1000 indicates that the queries are accessing large number of rows, the next step is to review high read queries. If you also see high Avg. Wait time, then it’s an indication of blocking, then review the blocking script output.  
Reference: -   
  
Object: - SQL Server:Locks   
Counter: - Average Wait Time (ms)  
Preferred Value: - < 500  
Description: - This is the average wait time in milliseconds to acquire a lock. Lower the value the better it is. If the value goes higher then 500, there may be blocking going on; we need to run blocker script to identify blocking.  
Reference: -