**Performance monitoring counters.**

**Here is a list of the most important counters to monitor in your IIS server:**

**Processor\% Processor Time**

This is the amount of time that the CPU is busy as opposed to waiting. In other words, it refers to the average percentage of processor time occupied

Whenever this value reaches 80% or more, then this indicates very high load on your server. It may be causes by High websites load or maybe other applications on your server.

You can check your application pool worker processor CPU consumption to check whether the websites or other applications are consuming your CPU.

**System\Processor Queue Length**

The processor queue is being filled up with threads and the processors are busy servicing other threads at the moment. If this counter is usually above 2 and the % Processor Time remains on high levels, then this is considered as a bottleneck.

**Memory\Available Mbytes**

Amount of physical memory that is free and can be used. If the free memory is equal or greater than 50%, the system is considered healthy. A value of 25% free memory indicates a potential problem.

If the free memory is less than 5%, the performance of the system is negatively impacted

**Memory\Pages/sec**

Amount of read and write requests from memory to disk.

Less than 500 Pages/sec are considered normal.

**Physical Disk\% Disk Time**

Percentage of time the disk was occupied, if this value is high then this will affect the performance of read and write on your disks which can cause low performance of your websites.

**Network Interface\Bytes Total/sec**

Amount of bytes – both sent and received – over the network.

Consider add network cards or upgrading to higher speed if this value is very high

**System\Context Switches/sec**

A context switch occurs when the kernel switches the processor from one thread to another—for example, when a thread with a higher priority than the running thread becomes ready. Context switching activity is important for several reasons. A program that monopolizes the processor lowers the rate of context switches because it does not allow much processor time for the other processes' threads.

(Microsoft)

High levels of context switching can occur when many threads share the same priority level. This often indicates that there are too many threads competing for the processors on the system. If you do not see much processor utilization and you see very low levels of context switching, it could indicate that threads are blocked)

* High context switches/sec – more than 5000 context switches per second
* Very high context switches/sec – more than 15,000 context switches per second

**Web Service:**

*You can view performance data for all applications or for specific ones, such as the Default Web Site or your specific Web Site instances.*

**Web Service\Bytes Total/sec**

Helpful to track and identify potential spikes in traffic.

**Web Service\Total Method Requests/sec**

Shows the number of all HTTP requests (counted since service startup).

**Web Service\Current Connections**

Shows the current number of connections established with the Web service.

**ASP.NET Applications\Requests/Sec**

Shows the throughput of the ASP.NET application on the server, if this value is very high then your application is considered as a heavy application, so you must consider monitoring and analyzing other counters to check it your resources need to be upgraded, if other counters are normal like CPU, Memory, or network. Then you have to check your application if it is working in bad behavior or if it really needs more resources.

**ASP.NET\Application Restarts**

If your application restarts are high, then this may indicate bugs in your application that you have to check.

**ASP.NET\Request Wait Time**

Shows the amount of time (in milliseconds) that the last request was held in the queue. It should be close to 0 ms

If it is greater than 1000 ms, the performance of the IIS server will be very low.

**ASP.NET\Requests Queued**

It’s like the processor queue but here its related only to the application.

Requests in queue are waiting to be processed. This counter should be monitored to find out when an application is overwhelmed. It must be as low as 2-3 requests.

**.NET CLR Exceptions\# of Exceps Thrown / sec**

Shows the number of exceptions per second that the application is throwing, obviously this must be low.

If high, then your application has many bugs and errors. And this can cause reboot of your application pools and very bad experience.

**.NET CLR Memory\# Total Committed Bytes**

Amount of virtual memory reserved for the application on the paging file (page file is the file on disk that memory uses when exceed its capacity) this must be very low and monitored with other factors to check where the problem is)

**Web Service\Get Requests/sec**

Amount of GET requests processed in a second. If the value is too high for a particular IIS server, then load balancing or clustering may be needed as a solution for large volume websites.

Source: Hassan Aboul hassn – Udemy IIS Course