

[Home](#) [Login](#) [Archive](#) [Submit News](#) [Downloads](#) [Reviews](#) [Articles](#) [About](#) [Stay Up2Date!](#)



Simplified Weld Monitor
Check key parameters for every weld. Easy to set up and operate.

www.weldaware.com

Accedian SLA-Meter™ PAA™

Constant in-service monitoring of SLA parameters at Layer 2/Layer 3

www.accionian.com

Tuning UNIX parameters
Automated

recommendations, graphs & explanations: AIX, Linux, Sol, HP/UX

www.sarcheck.com

4PSA Server Controller

Monitor the inside parameters of Unix/Linux servers from Windows.

www.4psa.com

End to End QoS and more

Automatic QoS for 600 applications Free Performance Survival Guide

www.packeteer.com

CREDITS

Thincomputing.net is maintained by Michel Roth, A Citrix Specialist from the Netherlands.

Thincomputing.net is sponsored by **Thin Computing Nederland**.



Lanmanserver and Lanmanworkstation Tuning

by **Michel Roth** on Tuesday 20 February 2007

Filesharing in Windows environments is usually of critical importance. After all, if you can not reach your files or have to wait 5 minutes every time you browse a share, the heat starts to build up in the IT department. Filesharing can easily become a bottleneck if not configured properly, especially in Terminal Server environments. In this article I'll explain what the important parameters are, what they do and how they relate to each other. Once you know this, you're able to tune your filesharing environments yourself.

Lanmanserver and Lanmanworkstation Tuning

Introduction

Filesharing in Windows environments is usually of critical importance. After all, if you can not reach your files or have to wait 5 minutes every time you browse a share, the heat starts to build up in the IT department.

File serving is more than just saving a file to your homedirectory of course. I wrote an two-part article on MSTerminalServices.org on file serving and Terminal server environments. I suggest you read that article ([part1](#) and [part2](#)) first to get a feel for the proper context of this article.

One of the main reasons I wrote that article is that filesharing can easily become a bottleneck if not configured properly, especially in Terminal Server environments.

To solve these performance problems, sometimes you have to tune the fileserver (lanmanserver) and the "fileserver-client" (lanmanworkstation). However this isn't for the faint of heart and can cause huge problems if you do it wrong. Unfortunately, documentation on these tuning parameters is rather scarce.

So in this article, I'll try to explain what the important parameters are, what they do and how they relate to each other. Once you know this, you're able to tune your filesharing environments yourself.

Please note, that there's also a great deal of optimisations that you can do in the "Terminal Server <-> Terminal Server Client" hemisphere. Although the basic filesharing principles also apply in that area, this article is not meant to help you perform those optimisations. Also, there's a lot of additional tweaking you can do in other parts of the (Terminal Server) registry. I'm aware of this. I've purposely left these optimizations out because I wanted this article to focus on performance of Filesharing components only.

This article was written assuming you're running Windows 2000 (SP4+) or Windows Server 2003, Service Pack 1.

Core Components

Before we go into deep, we need to take a look at the core components the Windows filesharing environment is made of. Filesharing in Windows is a classic example of a Client-Server mechanism. All you have to do become a file server is to check the box "file and printer sharing for Microsoft networks" in the network connection properties box. On the other end all you have to do to "use" this file server is to check the box "client for Microsoft Networks".

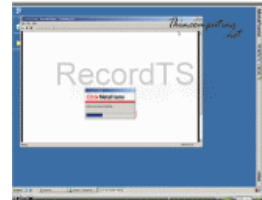
Both the server and the client components are run in a service. Not surprisingly this is the Server service for the server component and the Workstation service for the client components.

Settings for these services are stored in the Windows registry. For the Server service this location is:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanserver and the corresponding location for the Workstation service is
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\lanmanworkstation.

Lanmanserver

By default, the lanmanserver registry key on a freshly installed Windows Server 2003 Service Pack 1 machine looks like this:

THINCOMPUTING.NET PREMO



SEARCH

Search This Site

EVENT CALENDAR

Citrix Summit 2007
28 - 31 January 2007
Orlando, USA

BriForum USA 2007
23 - 25 April 2007
Chicago, USA

Pubforum Lisbon
May 2007
Lisbon, Portugal

Citrix iForum Global
22 - 25 October 2007
Las Vegas, USA

WELCOME

Username:

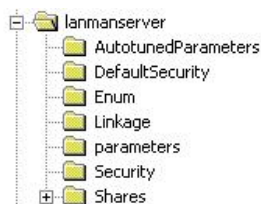
Password:

Login

☐ Remember me

[Signup]

[Forgot password?]



The first subkey we encounter is the "AutotunedParameters" key. If you check it, it will be empty. Don't worry; it's supposed to be empty. This registry key exists because, by default, the Server service is auto-tuning. This means that every time the system boots, the server takes a look at the hardware configuration and incorporates any changes in the configuration of the Server service. Changes in hardware that are monitored are the amount of memory and the number of processors. There's even a formula for it:

$$(4 * (MB * SMBServerPerfSetting) * OSVersion / 1) * (\#Processors)$$

where:

MB = Megabytes RAM on the server

SMBServerPerfSetting = .5 if "Minimize Memory Used"

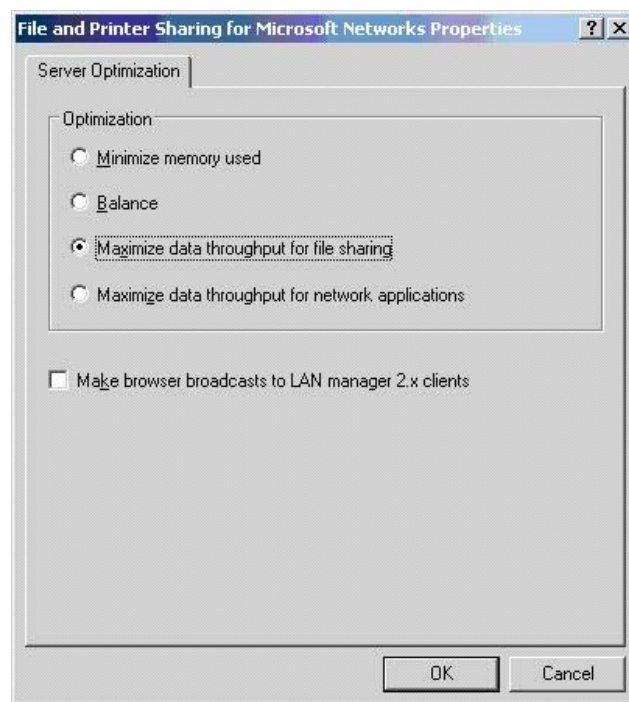
SMBServerPerfSetting = 1 if "Balance"

SMBServerPerfSetting = 2 if "Maximize Throughput for File Sharing"

OSVersion = 2 if running NTServer with > 16MB RAM

#Processors = is the number of processors in the system

In the formula you'll notice that it refers to the *SMBServerPerfSetting*. This brings us to the only GUI 'tool' native to Windows that you can use to "tune" the Server service. When you select the properties of a connection and then select the properties of "file and printer sharing for Microsoft networks", you should end up with a window like this:



This is where you can optimize the Server service for a specific role. Consequently, if you do the numbers you'll see that the higher you set the *SMBServerPerfSetting*, the higher the outcome of the formula is. But what is this number? Good question.

This number represents the value Windows will use for the *MaxWorkItems*, an important value in tuning the Server service. *MaxWorkItems* however, is just one of the parameters you can set to tune your fileserver. Let's take a look at the (registry) values.

Parameters

Before we begin discussing the relevant parameters you can use to tune the Server service you should know that you should create them in the *parameters* subkey of the *lanmanserver* registry key. Let's take a look at the most important parameters:

MaxWorkItems

As said, *MaxWorkItems* isn't alone in tuning the Server service. It is one of the most important parameters though. What does this parameter mean? Well, *MaxWorkItems* specifies the maximum number or work items (receive buffers for file requests) that the Server service is permitted to allocate at one time. If this limit is reached, then the transport must initiate flow control, which can significantly reduce performance or even deny new connections to the file server.

Possible values: 1-65535

InitWorkItems

This configures the number of work items allocated to a processor during startup. If this number is too low, it can significantly reduce performance or even deny new connections to the file server.

Possible values: 1-512

MaxMpxCt

This parameter permits a fileserver to provide a suggested maximum number of simultaneous outstanding client requests to itself. During negotiation of the Server Message Block dialect on this initial connection, this value is passed to the client's redirector where the limit on outstanding requests is enforced. A higher value can increase server performance, but requires more use of server work items (MaxWorkItems).

Possible values: 1-65535

MaxWorkItems and MaxMpxCt Relationship

The value for MaxWorkItems must be at least four times as large as that for MaxMpxCt. For example, if MaxMpxCt has a value of 4096, then MaxWorkItems needs to have a value of at least 16384.

MaxRawWorkItems

This value determines the maximum number of raw receive buffers that a server can allocate. If this limit is reached, server performance may be degraded.

Possible values: 1-512

MaxFreeConnections

This value controls the number of free connection blocks that are maintained for each endpoint.

Possible values: 2-4096

MinFreeConnections

This value specifies the minimum number of free connection blocks to be maintained for each endpoint. This setting can sometimes dramatically improve performance.

Possible values: 0-256

SizReqBuf

Specifies the size of the request buffers that the Server service uses. Small buffers use less memory, but large buffers can improve performance.

When running applications that use a lot of copy or move functions to a remote server (profiles anyone?), the speed at which this function completes is determined by network speed (of course) and by the SMB size. By increasing this block size, you will allow the server to complete its file copies faster. This will increase the performance of the application making the copy/move calls.

For computers running Windows Server 2003 and with 512 MB or more of physical memory, the default size of the request buffers is 16,644 bytes; for servers with less physical memory, the default size is 4,356 bytes. If this entry is present in the registry, its value overrides the default value.

Possible values: 1-65535

Lanmanworkstation

This key is where all the configuration data for the Workstation service is stored. The lanmanworkstation key by default, looks like this on a freshly installed Windows Server 2003 Service Pack 1 machine:



As you can see, there's no "autotunedparameters" here. However there is a parameters subkey in which we can do some tuning. Especially in Terminal server environments it is not uncommon to have to tune the Workstation service to alleviate performance problems. This is due to the nature of Terminal Servers. My article on MSTerminalServices.org discusses this in detail, but in a nutshell it's like this: the workstation service was (and is) designed for a single workstation (like your desktop). However, a Terminal Server can easily host 50 desktop sessions, but unless you do manually intervene this server mostly still is configured just as your desktop would. It's pretty obvious that this could lead to some performance problems.

Parameters

Although there aren't that many important parameters like in lanmanserver, there are still a few parameters of the Workstation service you should definitely know about.

MaxCmds

Specifies the maximum number of network control blocks that the redirector can reserve. The value of this entry coincides with the number of execution threads that can be outstanding simultaneously. Increase this value to improve network throughput, especially if you are running applications that perform more than 15 operations simultaneously.

So, MaxCmds actually serves the same purpose as the MaxMpxCt on the Fileserver. Not surprisingly these two parameters have a special relationship. It's like this: whenever a SMB session is set up (i.e. a shared file is accessed) the SMB session is negotiated. During this negotiation the Fileserver passes down the value of MaxMpxCt to the client (a Terminal server for example). The client then compares this value to his own MaxCmds value. The

lower of the two values then is used to set a maximum on the number of outstanding client requests to the File server.

Possible values: 1-65535

MaxThreads

The MaxThreads specifies how many threads is allowed to run at once, each thread allows one outstanding operation. By increasing this you can increase the amount of simultaneous work. Each extra execution thread will take 1 Kbyte of additional nonpaged pool memory.

Possible values: 1-255

MaxCollectionCount

Specifies the amount of data that must be present in the buffer of the redirector to trigger a write operation. If the amount of data in the buffer meets or exceeds this value, then it is written immediately. Otherwise, it is retained in the buffer until either more data is added or the value of the CollectionTime entry expires.

Possible values: 1-65535

Monitoring

Problems stemming from poor fileserving performance can sometimes be a bit tricky to pinpoint. One way to make sure is by using good ol' perfmon. The problem with interpreting perfmon counters is that you can never know what the "right" value is unless you have baselines your environment properly. So, what to monitor and how to interpret the values being monitor, is entirely up to you. However, there are some counters you can monitor that I can give some basic tips on. Configure perfmon to monitor the following counters:

Physical Disk

You can measure this on the Terminal Server as well, but you should start at the fileserver. If the queue length is more than 1 during a sustained period of time, then you're disks are hyperventilating. Give them some air: up your I/O throughput. Look on the software-side: are you paging a lot? (that'll kill your I/O throughput right there) or is your system disk heavily fragmented? Or on the hardware side: buy faster disks (15K SCSI) or upgrade your RAID controller.

Redirector

This is something you should only measure on your Terminal Server(s). You should monitor the "current commands" in the Redirector object. If the value is higher than 20 during sustained periods of time then you could have a bottleneck.

Server Work Queues

The Server Work Queues object should be monitored on the File server. You should monitor the "Available WorkItems" counter. Sustained values smaller than ten means that the File server is running out of work items. When it does, performance really starts to plummet. Make sure this doesn't happen by upping the *MinFreeworkItems* value (covered in the Thincomputing.net ADM template).

Server

In this object there's a counter called "Work Item Shortages". This value represents the number of times no work item was available or couldn't be allocated to service a "file request." Obviously, if you see any other value than zero, you need to start worrying. Uping the *InitWorkItems* or *MaxWorkItems* could help you out here.

Again, there's so much more you can monitor but interpreting the results depends heavily on your environment. Just browsing the performance monitor objects I mentioned and playing around with it will give you a lot more information.

Tuning

So what do I set these registry values to? Unfortunately it's not that simple. For starters, it depends on your specific environment and secondly a common denominator of almost every one of these registry values is that, when they are increased, they consume more kernel memory. Seeing as (the lack of kernel) memory often is a bottleneck in scaling up in Terminal Server environments, you should be very careful in adjusting/creating the registry settings we discussed. If you are not careful, you could end up with having more performance problems than you started out with. You need to know why.

Tuning LanManServer and LanManWorkstation in the registry, requires the use of more Non-Paged Pool memory. This can be a real issue on the File Server (LanManServer). Let me briefly explain where Non-Paged pool memory fits into the whole "2GB-Kernel-Memory-Bottleneck-Of-32-Bit-Windows".

When you have a 32 bit operating system, this means that you have a 32 bit address space. That translates to 4GB of addressable memory space (2 to the power of 32). This 4GB is evenly shared between the user mode and kernel mode. User mode is the memory space that applications run in and kernel mode is used by the system for everything else. This 2 GB kernel mode memory is divided into several areas, amongst which is the NonPaged Pool. Because there's only 2GB to share, the NonPaged pool gets configured with a maximum size at boot time. By default this is 256 MB. This 256 MB is the area in which you should perform your (LanManServer) tuning.

Why should you worry about this 256 MB? Well, because if the NonPaged pool is depleted then your system usually becomes unresponsive until some NonPaged pool becomes available again. So how does this apply to LanManServer tuning? Well, if you tune LanManServer in such a way that it allocates memory than the NonPaged pool has available and you indeed use up ALL of that allocated memory then you have effectively pushed Windows beyond its limits.

So what should you do? A safe way of doing it is to tune LanManServer in such a way that it can never deplete the NonPaged pool. The amount of memory LanManServer allocates in the NonPaged Pool is primarily determined by two parameters: *MaxWorkItems* and *SizReqBuf*. So if you set *MaxWorkItems* to 8192 and *SizReqBuf* to 16644 (default) (which in

reality is 20480 due to tracking overhead) the amount of memory LanManServer will allocate is (8192 x 20480 bytes) 160 MB. This fits nicely into the 256 MB NonPaged Pool area.

So it basically boils down to this: If you have more than 512 MB of memory in your Terminal Server (which is every Terminal Server on earth and adjacent planets) then SizReqBuf starts out at 16644. This allows you to push the MaxWorkItems value to 8192. If you try higher numbers to create more of these similar sized WorkItems AND your File Servers tries to use these, you run the chance of running out of NonPaged Pool.

So there is however a decent chance that having 8192 WorkItems does not cut it for you. This is when the bits start to hit the fan. If you're in that rather sad place, you really have only three options, with option 3 being the safest choice:

Try making the size of the WorkItems smaller (through the SizeReqBuf parameter) so you can safely set higher MaxWorkItems values. For example: If you set SizReqBuf lower to 8322 (plus a overhead of 3836 makes 12158 bytes) then this would allow you to have 13800 WorkItems (160MB / 12158 bytes).

You could even try to up the MaxWorkItems and SizeReqBuf values further with the risk that you run out of NonPagedPool. Now, you should also know that you can tune the Kernel Mode memory in such a way that more memory is allocated to the NonPagedPool. The downside to this is of course is that this memory is taken away from other parts of the Kernel Mode memory. I wouldn't go there if I were you (unless you're up there with the likes of Mark Russinovich).

Make sure that less Work Items are demanded from the File Server. This is a topic on its own but quick suggestions are: limit folder redirection (especially Application Data) or / and distribute File Services (put for example home directories on one Fileserver and redirected folders on another).

I have provided two .adm templates, one for lanmanserver and one for lanmanworkstation. I've separated these purposely because the lanmanserver adm template should be applied to your File Server and the lanmanworkstation adm template should be applied to your Terminal Servers.

[Thincomputing.net Lanmanserver Tuning.zip](#)

This template contains all of the discussed Lanmanserver parameters in this article. When you import the ADM template and enable the policy, it will set the following parameters to the Windows allowed maximum values:

- MaxWorkItems
- InitWokrItems
- MaxMpxCt
- MaxRawWorkItems
- MaxFreeConnections
- MinFreeConnections
- SizReqBuf

These optimisations should applied to your FILESERVER, not your Terminal Server. I've included the possibility to 'undo' the optimisations made the template. You can do this by selecting -Undo Lanmanserver Optimisations- and REBOOTING.

[Thincomputing.net Lanmanworkstation Tuning.zip](#)

This template contains all of the discussed Lanmanworkstation parameters in this article. When you import the ADM template and enable the policy, it will set the following parameters to the Windows allowed maximum values:

- MaxThreads
- MaxCollectionCount
- MaxCmds

These optimisations should applied to your TERMINAL SERVER, not your File server. I've included the possibility to 'undo' the optimisations made the template. You can do this by selecting -Undo Lanmanworkstation Optimisations- and REBOOTING.

Final Thoughts

Although some settings have been improved in Windows 2000 and even more in Windows Server 2003, I must say that I'm a bit disappointed that fileserving problems like I discussed in the article are still quite common in Terminal Server environments. These problems have been around just as long as Terminal Server has and one would think these problems would at least be a lot less common, but maybe that's just my point of view.

Microsoft, finally, recently has published an excellent article which discusses these issues in very good detail. This article isn't just about Terminal Server environments but it is still the best article Microsoft has ever written on the subject. Bookmark [KB317249](#).

I hope that this document has provided you with enough knowledge to combat file serving performance problems.

There's however a good chance that these problems with the file serving components of Windows will relatively soon be something of the past or at least be a lot less common. Windows Vista and Longhorn server will incorporate many changes, amongst which are major revisions in the file serving components. For example Vista comes with a major revision of the SMB protocol identified as SMB 2.0. The current protocol (SMB 1.0) was built to support file-serving solutions a couple of decades ago and was based on the assumptions existing then.

These are some of the key enhancements in SMB 2.0:

- SMB 2.0 supports an arbitrary, extensible way of compounding operations to reduce round trips. This makes the protocol less chatty as compared to SMB 1.0. Chattiness of SMB 1.0 has often been a major pain point.
- SMB 2.0 supports much larger buffer sizes compared to SMB 1.0.
- SMB 2.0 greatly grows the restrictive constants in the protocol, so we never need to worry about the protocol itself being the limiting factor for scalability. This includes increasing the number of concurrent open file handles on the server, and the number of shares that a server can share out, among other things.

- SMB 2.0 supports durable handles that can withstand short network glitches.

All these enhancements in SMB 2.0 will result in better performance and security over LAN and WAN.

Sounds good huh? I'll believe it, when I see it but the file serving future looks bright!

[>> List articles in this category](#)
[<< Back to articles front page](#)

Rating

Not rated

Render time: 0.5563 second(s).