NetTune Advisor - Optimizing Your System

Main Menu

The NetTune Advisor has been designed to simplify the choices in dealing with the many NetWare SET parameters. NetTune and its SmartTune component eases the effort of truly optimizing performance of a NetWare file server. This help file provides background on parameters that SmartTune monitors and manages. You also can work with these parameters either manually or by using the TimeTune feature to achieve the best server performance.

Parameter Analysis

SmartTune deals with the few parameters in NetWare that have the most significant performance impact. The following sections describe these parameters and the effect each parameter has on the system. You may choose to let SmartTune do all the work, or following the advisors explanations, modify the parameters yourself manually or with TimeTune.

Suggested Settings

Because Novell file servers are all very different in both make up and work load, precise parameter settings cannot be recommended for all situations. The recommended settings in the following sections are based on internal testing and data acquired from Novell.

Optimizing Backup

The following adjustments can significantly improve backup performance on most systems. The exact benefit to your system will depend to a great extent on where the backup takes place (server or workstation) and the memory/disk/CPU type. Because of delays caused by network topology, a server based backup will benefit most, and a workstation least. A SCSI disk system with bus master controllers will benefit significantly, as will a 386/486SX type system. A high performance EISA type system or multiple processor (NetFrame, Tricord with dedicated IOP) will show less improvement because such a system well exceeds the capability of most backup systems.

Dirty Directory Cache Delay Time - Increase to 10 Seconds

Dirty Disk Cache Delay Time - Increase to 10 Seconds

Immediate Purge of Deleted Files - Set to ON

Turbo Fat Re-Use Wait Time - Increase to 1 Hour

Dirty Cache Allocation Wait Time - Increase to 2 Minutes

Directory Cache Buffer Non-Referenced Delay Time - Increase to 3.75 Minutes

Maximum Directory Cache Buffers - Increase to 2000

Maximum Concurrent Disk Cache Writes - Decrease to 20

Maximum Concurrent Directory Cache Writes - Decrease to 5

Theory behind the adjustments

Backup is primarily a serial non-redundant read process. Therefore optimizing should focus on the parameter changes favorable to this set of activities. The number of disk writes during a backup is very small compared to the number of reads. The default NetWare settings should be adjusted to favor the read-intensive environment.

Optimizing Database Activity

Optimizing your database activity is similar to optimizing backups, with minor variances. Most database activity is read intensive. The assumption here is that database activity is occurring in a transactional, client-based operation such as using dBase or Clipper based application code, with the program and data at the server. In limited cases these changes have improved SQL type operations.

Dirty Directory Cache Delay Time - Increase to 10 Seconds

Dirty Disk Cache Delay Time - Increase to 10 Seconds

Turbo Fat Re-Use Wait Time - Increase to 1 Hour

Directory Cache Allocation Wait Time - Increase to 2 Seconds

Directory Cache Buffer Non-Referenced Delay Time - Increase to 5 Minutes

Maximum Directory Cache Buffers - Increase to 2000

Maximum Concurrent Disk Cache Writes - Decrease to 20

Maximum Concurrent Directory Cache Writes - Decrease to 5

Optimizing Transaction Response Time

Optimizing Transaction Response Time is similar to Database activity, but with a couple of minor changes. Please review the Database activity, and ask yourself, "do I perform more reads, writes, or a combination of both?" If you perform a balanced word processing/ spreadsheet/database load, review your History file as recorded by NetTune. Look under disk Reads vs. Writes to see if a pattern emerges.

Dirty Directory Cache Delay Time - Increase to 10 Seconds

Dirty Disk Cache Delay Time - Increase to 10 Seconds

Turbo Fat Re-Use Wait Time - Increase to 1 Hour

Directory Cache Allocation Wait Time - Increase to 2 Minutes

Directory Cache Buffer Non-Referenced Delay Time - Increase to 5 Minutes

Maximum Directory Cache Buffers - Increase to 2000

Maximum Concurrent Disk Cache Writes - Increase to 100

Maximum Concurrent Directory Cache Writes - Increase to 50

Perform an hourly and then a daily summary of transactions, investigating the nature of your drive activity. This data is then used to set the Maximum Concurrent Disk and Directory Cache Writes to the optimum value. If your system performs a balanced mix, set the values to 50 each. If you perform significantly more reads, use the Database parameters. If you perform a significant number of Writes, use the Transaction parameters indicated above. By setting the Maximum Concurrent disk and directory cache writes to 100 you are optimizing writes.

By setting the directory cache buffer non referenced delay time to the maximum 5 minute value, we are sacrificing some memory to increase transaction hit rates on cache. The goal of Transaction Response Time is to maximize the chances that data is in cache. This is different from our Database mix because we are assuming in Database activity that we are looking up/reading often but Writing little.

Reducing CPU Workload

Reducing CPU workload can be of significant value when dealing with a 386 or 486/SX system. These systems typically have a maximum of 16 MBytes of memory and are often "job servers".

Dirty Directory Cache Delay Time - Increase to 10 Seconds

Dirty Disk Cache Delay Time - Increase to 10 Seconds

Immediate Purge of Deleted Files - Set to ON

Turbo Fat Re-Use Wait Time - Increase to 1 Hour

Dirty Cache Allocation Wait Time - Increase to 2 Minutes

Directory Cache Buffer Non-Referenced Delay Time - Increase to 5 Minutes

Maximum Directory Cache Buffers - Leave at default of 500

Maximum Concurrent Disk Cache Writes - Increase to 100

Maximum Concurrent Directory Cache Writes - Increase to 50

By purging all deleted files, caching all available disk reads and writes, we are assuming this file server does little file activity and is instead being used for a specific purpose - print server, backup server, or communications server. Varying the Concurrent Disk and Directory Cache Writes mix between 50-100 will have little impact on CPU utilization, but will have an impact on Disk read/write speed. This parameter can be changed as needed to optimize performance.

Reducing Memory Constraints

On servers primarily used in a read only mode, the memory needed can be reduced.

This combination of parameters is useful on a server that has a relatively heavy work load, but is short of RAM relative to drive capacity. For example, a 16MB server with a 1GB hard drive may have only 1200 cache buffers free. Adding another 1GB drive will reduce cache buffers to 300 or so, putting a severe squeeze on performance. By using the following settings, you may gain some performance without 'breaking the RAM budget'.

Dirty Directory Cache Delay Time - Increase to 10 Seconds

Dirty Disk Cache Delay Time - Increase to 10 Seconds

Immediate Purge of Deleted Files - Set to ON

Turbo Fat Re-Use Wait Time - Decrease to 1 Minute

Dirty Cache Allocation Wait Time - Increase to 2 Minutes

Directory Cache Buffer Non-Referenced Delay Time - Increase to 3.75 Minutes

Maximum Directory Cache Buffers - Reduce to 100

Maximum Concurrent Disk Cache Writes - Reduce to 10

Maximum Concurrent Directory Cache Writes - Reduce to 5

These parameter settings limit the number of resources NetWare would normally allow to be allocated, reducing the amount of memory needed. Often reducing memory usage will cause an inversely proportional increase in server utilization.

You may also try reducing the number of file/record locks permitted by a connection, preventing over-use of resources by rogue workstations.

SmartTune Quick Analysis Parameters

The SmartTune Quick Analysis section of NetTune provides an opportunity to modify parameters that only need to be set once. These parameters include the following:

Auto TTS Backout Flag

Cache Buffer Size

Enable Disk Read After Write Verify

Immediate Purge Of Deleted Files

Maximum Physical Receive Packet Size

Minimum Packet Receive Buffers

SmartTune Automatic Tuning Parameters

The SmartTune Automatic Tuning of NetTune works with parameters that can be modified on the fly to improve performance. The SmartTune Artificial Intelligence Engine monitors internal operating system conditions and may adjust one or more of the parameters listed in this section. SmartTune can tune in six different categories which can individually be activated or deactivated.

Optimizing File Cache

Maximum Concurrent Disk Cache Writes

Optimizing Disk I/O

Dirty Disk Cache Delay Time

Dirty Directory Cache Delay Time

Optimizing Directory Cache

Directory Cache Allocation Wait Time

Directory Cache Buffer NonReferenced Delay

Maximum Concurrent Directory Cache Writes

Maximum Directory Cache Buffers

Minimum Directory Cache Buffers

Optimizing Packet Receive Buffers

Maximum Packet Receive Buffers

New Packet Receive Buffer Wait Time

Optimizing Service Processes

Maximum Service Processes

New Service Process Wait Time

Optimizing Turbo-FAT

Turbo FAT Re-Use Wait Time

Auto TTS Backout Flag

Default: OFF

This parameter can only be set in the STARTUP.NCF file.

NetTune contains a "Down Server" menu item that can be used to remotely down and restart a server. If the server AUOTEXEC.BAT file has a loop to restart the SERVER.EXE, then when the server is downed, it will restart. This convenient feature allows you to make changes to a server's NCF files and then restart it, without the need to be at the server system console. By setting this flag to ON, prompts normally presented at the system console are bypassed, allowing the server to restart without human intervention. While this may not seem like a performance related parameter, it is definitely a time saving feature.

Cache Buffer Size

Default: 4096

Range: 4096 to 16384

This parameter can only be set in the STARTUP.NCF file.

NetWare allows variable size for its cache blocking. This provides a possible opportunity to improve both utilization and user response times. The cache buffer size is determined at startup. The volume block size is set when the volume is created. The system is optimized when the volume block size and the cache buffer size are the same. In a system with multiple volumes, each with a different block size, this will not be possible. In this case, NetWare requires that the cache buffer size be set to the same as the smallest configured volume block size.

These is some advantage to a larger block size. Large volume block sizes provide a more efficient use of the disk, and also reduces the size of the FAT which is always cached to memory. The tradeoff however is that more disk space may be wasted if the file system holds many files that are much smaller in size that 8KB.

Valid cache buffer sizes are 4096, 8192 and 16384 for NetWare 3.x.

Enable Disk Read After Write Verify

Default: ON

The NetWare operating system verifies that a block written to disk can be read by reading the block after a write. This verification goes hand in hand with the Hot Fix function. The default setting of ON allows the system to prevent a file from being written to a bad disk block by attempting to read it after it is written. If a problem is encountered, the Hot Fix procedure is invoked and the data is re-directed to a good block. This procedure is implemented in software.

Many of today's controllers and hard drives perform this verification at the hardware level, introducing an unnecessary level of redundancy. If your hard drives/controllers use hardware read-after-write verification, set this parameter to OFF. If the hardware detects a bad block during a write, the Hot Fix function will still be invoked by the disk driver.
This parameter is nothing more than a global variable available to the disk driver. However, most drivers don't check the variable dynamically. So although this parameters defaults to ON, most drivers may actually ignore the setting.

You can use MONITOR to determine what type of verification your disk drivers are using. There are three levels of verification supported: software level, hardware level or none. If MONITOR shows that hardware level is being used, this is the most efficient and should be left alone. If however MONITOR shows that software level is being used, you may be able to change to the more efficient hardware level. (If this option is not available, check with your disk driver developer for a later version of the driver that may have the option.

If reliability is more important that performance, in your environment, you may choose to leave this parameter in the ON state.

This parameter should be left to ON for older subsystems that do not have Read-After-Write Verify logic built into the hardware.

Immediate Purge Of Deleted Files

Default: OFF

This SET parameter turns on and off the NetWare salvageable file system. When this parameter is set to OFF, the salvageable file system uses excess free space to store deleted files. You can recover files that have been deleted by mistake, hours later or even longer. Many administrators have and will rely on this feature. The performance trade-off is the system overhead required to keep track of the deleted files.

Files which were deleted while the flag was OFF, will remain salvageable, even if the flag is turned off. They will remain salvageable until the space they occupy is needed, or the deleted files are purged.

If you perform nightly backups and have adequate access to that data, you may choose to purge all deleted files after your backup routine, and set this parameter to ON.

Maximum Physical Receive Packet Size

Default: 1514

Range: 618 to 4202 (NetWare 3.11)

Range: 618 to 24682 (NetWare 3.12)

This parameter can only be set in the STARTUP.NCF file.

Communications buffers are allocated from permanent memory, which is obtained from the file cache memory. Once permanent memory has been allocated, it is never returned to the file cache until the server reboot.

This parameter should be set no higher that the largest maximum packet size of each LAN card (+56 bytes) to insure that the largest buffer size can be accommodated without wasting memory. The following are the maximum packet sizes for various network physical layers:

Default Ethernet	1,514
ARCNET	618
Ethernet	1,130
Token Ring (4MB)	2,154
Token Ring (16MB)	4,202

Minimum Packet Receive Buffers

Default: 100 (NetWare 3.11)

Default: 400 (NetWare 3.12)

Range: 50 to 2000

This parameter can only be set in the STARTUP.NCF file.

This parameter can affect performance when lots of buffers are required immediately, but are not available. The required buffers will be dynamically allocated by NetWare if the number in use reaches or exceeds 75%, up to the MAXIMUM, but the server response will be sluggish until then. A higher value at startup can prevent this initial slowdown.

A general rule of thumb: set the initial value to one buffer for each 5 workstations, plus four buffers for each LAN card. Bus master LAN cards may require a higher number.

Maximum Concurrent Disk Cache Writes

Default: 50

Range: 10-100 (NetWare 3.11)

Range: 10-1000 (NetWare 3.12)

This parameter allows you to control the flow of writes to the hard drive. Increasing the value causes writes to be serviced more efficiently. Decreasing the value causes reads to be serviced more efficiently.
NetTune and MONITOR show the pending number of writes and the number of dirty cache buffers (cache buffers that have been modified but not written to disk). A write intensive environment will cause the number of dirty cache buffers to rise. When 75% of total cache buffers are dirty, NetWare will hold off file writes. This would definitely indicate a disk related bottleneck and would manifest itself by reduced or sluggish performance. By increasing this parameter, the number of buffers which can be written to disk is increased and performance should improve. If performance does not improve after increasing this value, this could indicate disk channel saturation.

Dirty Disk Cache Delay Time

Default: 3.3 seconds

Range: 0.1 to 10 seconds

This parameter can be used to decrease the probability of redundant disk writes. When NetWare receives a write request, the cache block is modified but not immediately written to disk (the block is marked as dirty). If NetWare performed an immediate write to disk, this would create a 1:1 ratio of physical writes to write requests. By waiting the time specified by this parameter before writing the block, NetWare increases the probability that another write to the same block will occur, and both write requests could be satisfied with one physical write.

In environments where applications make many small write requests, increasing this value can improve performance. Decreasing this value will result in more frequent disk I/O.

Directory Cache Allocation Wait Time

Default: 2.2 seconds

Range: 0.5 seconds to 2 minutes

This parameter controls the rate of growth of directory cache buffers. It is the minimum amount of time to wait between new directory cache allocations. Until this time expires all new requests for a directory cacne buffer are ignored.

If the wait time is too low, peak usage requests will cause more resources to be allocated to directory caching than needed.

If the wait time is too high, the operating system will be slow in allocating the directory buffers necessary to service the usual number of directory requests.

If directory searches seem slow after the file server has been running, you may want to decrease this time.

Directory Cache Buffer NonReferenced Delay

Default: 5.5 seconds

Range: 1 second to 5 minutes

This parameter controls the amount of time that must pass before a non-referenced directory cache buffer is re-used. If all directory cache buffers are referenced within this delay time, a new directory cache buffer is allocated using the directory cache allocation wait time.

If you increase this parameter, the operating system allocates more directory cache buffers. The longer time quickens directory access because a directory is more likely to be cached in memory.

If you decrease this parameter, directory access time can slow down because a directory is less likely to be in memory. But decreasing the time reduces the need for directory cache buffers since the operating system waits less time before it reuses a buffer.

This parameter has less impact on performance if the directory cache is appropriately sized to the workload.

Dirty Directory Cache Delay Time

Default: 0.5 seconds

Range: 0.0 to 10 seconds

This parameter can be used to decrease the probability of redundant directory cache writes. When NetWare receives a directory request (write), the cache block is modified but not immediately written to disk (the block is marked as dirty). If NetWare performed an immediate write to disk, this would create a 1:1 ratio of physical writes to write requests. By waiting the time specified by this parameter before writing the block, NetWare increases the probability that another directory write to the same block will occur, and both write requests could be satisfied with one physical write.

In environments requiring a large directory cache, increasing this value can improve performance. Decreasing this value will result in more frequent disk I/O.

Maximum Concurrent Directory Cache Writes

Default: 10

Range: 5 to 50

This parameter allows you to control the flow of directory writes to the hard drive. Increasing the value causes writes to be serviced more efficiently. Decreasing the value causes reads to be serviced more efficiently. Increasing the value causes writes to be serviced more efficiently.

If your system has a large directory cache with thousands of frequently accessed small files, and your disk subsystem is capable of a high rate of requests per second, you may benefit from increasing this value.

Maximum Directory Cache Buffers

Default: 500

Range: 20 to 4000

This parameter sets the upper limit to the growth of directory cache buffers. When NetWare allocates a directory cache buffer, the allocation is permanent until the server is rebooted. This parameter keeps the operating system from allocating too many directory cache buffers.

Increase this limit if the file server responds slowly to directory searches.

Decrease this limit if too much memory is being allocated for directory caching. If users are alerted that the server is low on memory, this parameter should be the first one to be reduced. However you must reboot the server to return the memory to the cache buffer pool.

Minimum Directory Cache Buffers

Default: 20

Range: 10 to 2000

This parameter determines the minimum number of cache buffers that the operating system can allocate for directory caching. This parameter needs to be high enough that directory searches can be done quickly, but no higher than necessary.

If NetWare does not need the minimum number of directory cache buffers, the buffers cannot be reallocated to file caching. The unneeded portion remains unused.

If the file server responds slowly to directory searches after it has been rebooted, monitor the number of directory cache buffers usually allocated for directory caching. If the number is significantly higher than this limit, consider increasing the limit to remove the delay time that would normally occur while the server is self-configuring.

Maximum Packet Receive Buffers

Default: 400

Range: 50 to 2000

This parameter controls the maximum number of packet receive buffers that the operating system can allocate.

If the number of packet receive buffers is at its maximum, increase this value in increments of 10 until you have one packet receive buffer per workstation.

If you have EISA or microchannel bus masters boards in your file server, increase this parameter to allow for at least 5 buffers per board. If you are receiving "No ECB Available" erros, increase the parameter to allow for 10 packet receive buffers per board.

If the number of allocated service processes is at its maximum, you can increase the Maximum Number Of Service Processes parameter to decrease the need for more packet receive buffers.

New Packet Receive Buffer Wait Time

Default: 0.1 seconds

Range: 0.1 seconds to 20 seconds

This parameter controls the rate of growth of packet receive buffers. It is the minimum amount of time to wait between new receive buffer allocations.

This parameter prevents the operating system from granting too many packet receive buffers during a sudden peak inusage. If you have a EISA bus master board in your file server, this parameter should not be changed.

Maximum Service Processes

Default: 20

Range: 5 to 40

This parameter controls the maximum number of service processes that can be created for servicing client NCP requests. A service process is a thread that acts upon incoming service requests. The minimum number of service processes is not a parameter, but does exist and has a value of 10. This means that NetWare will freely allocate up to 10 service processes before it starts using the wait time parameter.

In many cases the number of service processes may remain low. There are several cases when you may want to increase this parameter though. Servers servicing many users 200 or more, or from workloads involving imaging, audio and video. Increasing the number of service processes may reduce the need for allocating more packet receive buffers.

Whenever you increase this number, make sure that the number of service processes remains two or three below the maximum.

New Service Process Wait Time

Default: 2.2 seconds

Range: 0.3 seconds to 20 seconds

This parameter controls the rate of growth of service processes. It is the minimum amount of time to wait between service process creations.

Turbo FAT Re-Use Wait Time

Default: 5 minutes 29.6 seconds

Range: 0.3 seconds to 1 hour 5 minutes 54.6 seconds

NetWare, in order to improve performance in accessing large files, will automatically create a Turbo-FAT entry for files with more that 64 FAT entries (such as large or gragmented files). Because of the overhead in both CPU and memory, this Turbo-FAT index is not deleted when the file is closed. This parameter controls how long the Turbo-FAT is kept around after tile is closed. If your environment frequently opens and closes very large files (databases?), you may improve performance by increasing the value of this parameter to reflect a time greater than your file access frequency.