# Novell Technical Information Document

**Performance, Tuning and Optimization - Part 1** *(Last modified: 18MAY2000)*

This document (**2943356**) *is provided subject to the disclaimer at the end of this document.*

**Issue**

TID2943356 and TID2943472 cover the various areas to optimize your server's performance. It also looks into areas of pro-active preventive maintenance on your server and how to achieve the best results. These actions taken also prevent the possibility of server abends and crashes.

The following areas are covered in TID2943356 and TID2943472:

1. MONITOR.NLM and Utilization
2. Operating System Patches and Other NLM updates
3. NDS and DS.NLM
4. Service Processes
5. Upgrade Low Priority Threads
6. LRU Sitting Time
7. Physical Packet Receive Buffers
8. Packet Receive Buffers
9. Packet Burst
10. Cache Buffers
11. Dirty Cache Buffers and Current Disk Requests
12. Host Bus Adapters, Mirroring and Duplexing
13. Directory Caching
14. Block Suballocation
15. Disk Block Size
16. Hotfix and Redirected Blocks
17. Suballocation and the Number of Free Blocks and Disk Space for Each Volume
18. File Compression
19. File Decompression
20. Read After Write Verification
21. Garbage Collection
22. Read/Write Fault Emulation and Invalid Pointers
23. Interrupt Allocation and Configuration
24. Set Reserved Buffers Below 16 Meg
25. AppleTalk
26. Printing

1. MONITOR.NLM and Utilization

The Monitor utilization is not an entirely accurate number. Some server processes call a NetWare function called CyieldWithDelay or CyieldUntilIdle. If there is a thread spinning on one of these functions, the server will appear to have high utilization. Utilization will then be inaccurate. Don't panic when utilization goes to 100 percent for a few seconds and bounces down. This is normal for all 4.10 servers.

2. Operating System Patches and Other NLM updates

For 4.10, apply the current patches from 410PTx.EXE (currently 410pt8b.exe). These patches fix problems that resulted in high utilization and other issues. These patches also fix a variety of additional problems. Loading all of the patches will reduce your chances of having future problems.

For IntranetWare or NetWare 4.11, use the support packs by the file name NW4SPx.EXE. For NetWare 5, use the NW5 support packs by the file name NW5SPx.EXE. Support packs contains not only OS patches but all the necessary updates and enhancements to the server NLMs. (Current version, at the last update of this TID, of NW 4.10 patches is 410PT8B.EXE, NW 4.11 and NW 4.2 support pack is NW4SP7A.EXE and NW 5.0 support pack is NW5SP3A.EXE).

OS patches fixes issues with the SERVER.NLM or LOADER.EXE. There are also other fixes in new updates to NLMs released (e.g. CLIB.NLM) which resolves other issues, that can pertain to high utilization. The new updates also adds enhancements to the current versions. It is advised to always apply the latest updates to your server as well.

Please refer to support.novell.com web page for the latest patches and updates. It is recommended to apply all the latest patches and updates specified in the minimum patch list found in http://support.novell.com/misc/patlst.htm.

Also, apply the latest LAN and Disk drivers from your hardware vendor.

3. NDS and DS.NLM

Update DS to version 4.89a or later for NW4.10. Novell Support has seen issues with DS causing high utilization and 4.89a has solved all these issues. (Current version, at the last update of this TID, of DS.NLM for NW 4.10 is 5.18 from DS410Q.EXE, NW 4.11 and NW 4.2 is 6.09 from DS411S.EXE, and NW 5 is 7.40 from DS7A.EXE)

Efficient tree design, partitioning and replication is essential to avoid utilization problems. The size, type and number of partition replicas can cause utilization problems if not managed properly. Check also the total number of NDS objects residing in the partitions of that server. DS needs to keep synchronization among all servers in the replica ring. The more replicas there are of any partition, the more traffic will be on the wire. Novell recommends having at least three replicas of each partition in the tree. This provides fault tolerance and allows for DS recovery if a database were to become corrupt.

4. Service Processes

Service processes are threads of execution that act as hosts to incoming service requests. NetWare 4.X is capable of allocating up to 1000 service processes. Set maximum service processes to 2-3 per connection. It is recommended to set the Maximum Service Processes to 1000 (the maximum number allowed). If the server does not require an additional service process, it will not allocate. You can also set new service process wait time to 0.3 (default is 2.2).

Refer to HIGHUTL1.EXE for more info on Service Processes and also on how to immediately increase the number of service processes to your desired value.

5. Upgrade Low Priority Threads

Verify that SET UPGRADE LOW PRIORITY THREADS is set to OFF. If it is ON it will contribute to any utilization problems the server may be having. This parameter does not apply in NW5 servers.

6. LRU Sitting Time

NetWare's file caching subsystem is a pool or collection of 4 KB memory pages. After loading the OS, system NLMs, and application NLMs, NetWare initializes all remaining memory as the file cache pool. At the beginning of the list is the "list head," where new cache buffers are inserted into the list. The end of the list is the "list tail," where old cache buffers are removed from the list. Each cache buffer in the list is linked to the next cache buffer, and each one includes a time stamp indicating the time the cache buffer was inserted into the list head.

When the server receives a disk I/O request for data that is not currently in cache (a cache "miss"), the data is read from the disk and written into one or more cache buffers that are removed from the list tail. Each newly filled cache buffer is time-stamped with the current time and linked into the list head. A newly filled cache buffer is designated as the most-recently-used (MRU) cache buffer because it has resided in cache for the least amount of time.

A cache "hit" a frequent event in NetWare environments occurs when a disk request received by the server can be serviced directly out of cache, rather than from disk. In this case, after the request is serviced the cache buffer containing the requested data is removed from the list, time-stamped with the current time, and relinked into the list head. In this manner, MRU cache buffers congregate at the head of the list. This characteristic of the list is important to understand, because you want your MRU cache buffers to remain cached in anticipation of repeated use and repeated cache hits.

At some point in this process, the file cache pool becomes full of recently used data. This is where the least-recently-used (LRU) cache buffer comes into play. LRU cache buffers are buffers that were originally filled from the disk, but haven?t been reused as frequently as the MRU cache buffers at the list head. Due to the relinking of MRU cache buffers into the list head, LRU cache buffers congregate at the list tail. When new cache buffers are needed for data requested from disk, NetWare removes the necessary number of LRU cache buffers from the list tail, fills them with newly requested data, time-stamps them with the current time, and relinks them into the list head.

The resulting NetWare file cache subsystem gives preference to repeatedly used data and holds onto less frequently used data only as long as the memory isnt needed for repeatedly used data.

When tuning file cache, then, the ideal scenario is one in which every repeated use of recently accessed data can be serviced out of cache. This is accomplished by sizing server memory so that the resulting file cache pool is large enough to retain all repeatedly used data.

The LRU Sitting Time statistic is calculated by taking the difference between the current time and the time stamp of the LRU cache block at the tail of the cache list. LRU Sitting Time measures the length of time it is taking for an MRU cache buffer at the list head to make its way down to the list tail, where it becomes the LRU cache buffer. One might refer to this measurement as the cache "churn rate" because, whether from cache hits or misses, every cache buffer in the list is being reused within that period of time. Check that the LRU Sitting Time does not go below 15 minutes. If it drops below 15 min, you may need to add more memory.

7. Physical Packet Receive Buffers

The Physical Receive Packet Size defines the size of the buffers that are available to LAN drivers and protocols to store packets. This parameter does not actually define the size of the physical packets on the wire. The only relation to the physical size is that of course the packet buffer should be big enough to
hold a complete packet of maximum size.

The Maximum Physical Receive Packet Size should be set according to the kind of network it is on. In most cases, this is 1524 bytes for Ethernet segments, 4540 bytes for Token-Rings and FDDI and 618 bytes for Arcnet and LocalTalk. These values are taken from the "Novell BorderManager Installation and Setup" manual, the chapter on "Installing Novell BorderManager", page 9. Certain products installed have specific requirements in which you will need to refer to your manuals for instructions. The most important thing to know is that there is no best general value for this parameter, even if you known that you are only working with Ethernet. The minimum value to use depends on:

a. the LAN drivers used
b. the protocol drivers used
c. in case of WAN connections, the MTU size on the WAN and the WAN drivers used.

It is a common mistake for people to think that for Ethernet, the value 1514 should be enough because Ethernet packets are no bigger than 1514 bytes. This is wrong. The problem is that the drivers may use the buffers for other things than just storing the data in it. They may store extra information in the buffer (like for
example scatter/gather lists for busmaster adapters). They could also align data on preferred boundaries for better busmaster performance.

In case of drivers for Ethernet cards, the worst case known to me are the LAN drivers for Ethernet cards based on Intel 82557/8/9 chips. This does for example include E100B.LAN (Intel), IBMFE.LAN (IBM), HPTX.LAN (HP) and N100.LAN (Compaq) drivers. These drivers require at least 2048 buffers. With even one byte less of buffer size, you may experience problems ranging from bad performance to random untrackable abends up to drivers not working at all. In all cases, you will typically also see the "no ECB available" counter of the LAN driver increasing abnormally.

Given the wide distribution of cards based on these chips, I would not recommend anyone to use a "Maximum Physical Receive Packet Size" value lower than 2048, as you never know if such a card may be added later. At that moment, you will probably not think about changing the value, and so you will have a rather bad configuration.

Leaving the default value of 4224 (as of NW5) should always be safe. Reducing the buffer size will not fix an abend problem, unless you are very short on memory, in which case adding memory would be a better idea. The only benefit from changing the default value of 4224 is to safe a few hundred KB of memory (depends on the number of buffers you use).

8. Packet Receive Buffers

Receive buffers are used to store incoming packets from each of the networks attached to a NetWare server. A good rule of thumb for the minimum packet receive buffers setting is approximately 2-3 receive buffer for each connection and maximum packet receive buffers to 4000 (or any higher value).

Also, check the No ECB (Event control block) Available Count information in the LAN/WAN Information option in MONITOR.NLM. These messages indicate that the server was unable to acquire sufficient packet receive buffers, usually called event control blocks (ECBs). Running out of ECBs is not a fatal condition. Servers that run for several days where high loads occur in peaks might exceed the set maximum number of ECBs, causing the system to generate ECB system messages. If these situations are caused by occasional peaks in the memory demand, you should probably maintain your current maximum ECB allocation and allow the message to be generated at those times. On the other hand, if your server memory load is very high and you receive frequent ECB allocation errors, you should probably set your maximum ECB allocation higher. Use the following SET command in the AUTOEXEC.NCF file: SET MAXIMUM PACKET RECEIVE BUFFERS = number

Note that the memory allocated for ECBs cannot be used for other purposes. The minimum number of buffers available for the server can also be set in the STARTUP.NCF file with the following command: SET MINIMUM PACKET RECEIVE BUFFERS = number.

If the current Packet Receive Buffers rises above the minimum set level after the server has been up for a period of time, set the minimum Packet Receive Buffers to the current level.

9. Packet Burst

Novell support has seen high utilization problems caused by packet burst from both NetWare and Microsoft requester clients. Most of these problems have been solved with an OS patch or a new requester. Load the 410 OS patches and get the new FIO.VLM which fixes the problem with the Novell requester. The file is VLMFX3.EXE on the Internet. (Current version of VLM is 1.21)

Troubleshooting:
Novell technical support has a module called PBRSTOFF.NLM that will disable Packet Burst from the server. The module needs to

be loaded when the server comes up. Only connections that are established after the module is loaded will have packet burst disabled. This will isolate utilization problems related to packet burst.

10. Cache Buffers

When the server boots, all free memory is assigned to file caching. As demand increases for other resources (e.g. directory cache buffers), the number of available file cache buffers decreases. The operating system does not immediately allocate new resources when a request is received. It waits a specified amount of time to see if existing resources become available to service the demand. If resources become available, no new resources are allocated. If they do not become available within the time limit, new resources are allocated. The time limit ensures that sudden, infrequent peaks of server activity do not permanently allocate unneeded resources. The following parameters are dynamically configured by the operating system.

Directory cache buffers
Disk elevator size
File locks
Kernel processes
Kernel semaphores
Maximum number of open files
Memory for NLMs
Router/server advertising
Routing buffers
Service processes
TTS transactions
Turbo FAT index tables.

As a rough guideline, if the amount of cache buffers (checked from MONITOR.NLM, Resource utilization) drops below 40%, more memory should be added to the file server. This percentage rule has been commonly used for many years when the average system had 16MB to 32MB, and most systems nowadays have more cache buffers (in terms of percentage) available. An example is that 40% of 32MB is 12 MB, while 40% of 1024MB is 409 MB.

In most cases, impact to performance can be seen when the amount of cache buffers drops below 40%. The percentage value is calculated from having total cache buffers divided by the original cache buffers. The total cache buffers and original cache buffers can be found at the General Information screen of the MONITOR.NLM. Please refer to the March 1997 AppNotes on Optimizing IntranetWare Server Memory.

A brief note on fine tuning:
As resource (e.g. memory) on every server is limited (until more memory is added), the whole task of fine tuning a server is to find a balance between the limited resource with the server response to the workstation. Take an example of a server with only 64MB of memory. If I allocate more cache for directory cache buffers and packet receive buffers, this will reduce the amount of total cache buffers used for file caching. By over allocating the amount of other resources that consume memory would reduce the amount of memory for file caching, and hence slowing down the server rather than speeding it up. For this example, a balance of an adequate amount of directory cache buffers and packet receive buffers, giving enough total cache buffers for file caching, would give a better server performance than if there is insufficient directory cache buffers or too many directory cache buffers (taking into consideration the current limited resource on the server).

11. Dirty Cache Buffers and Current Disk Requests

If the Dirty Cache Buffers and Current Disk Requests (as seen from the MONITOR screen) is high, it means that that the number or dirty cache waiting to be written to disk, and the number of pending disk requests (reading from disk) is high.

You can set the Maximum Concurrent Directory Cache Writes (default of 10), the Maximum Concurrent Disk Cache Writes above the default value (default of 50). You can also set the Dirty Disk Cache Delay Time to a value below the default (default of 3.3 Sec).

You can adjust the settings till the values of dirty cache buffers and current disk requests goes to 0 periodically. The values should go in bursts and return to 0. If after making adjustments and the dirty cache buffers and current disk requests are still high, it would mean that your disks cannot handle the load. It would then be advisible to either change to faster disks or to split the server load up. Often times, the disk is one of the common bottlenecks.

You may like to start with the following settings and fine tune subsequently:

```
SET Maximum Concurrent Disk Cache Writes = 500
SET Dirty Disk Cache Delay Time = 0.5
SET Maximum Concurrent Directory Cache Writes = 100
```

12. Host Bus Adapters, Mirroring and Duplexing

While configuring multiple devices, it is advisible to put slow devices on a different channel or host bus adapter from the fast devices. Slow devices are like tape drives and CDROMs. Fast devices are like disk drives.

Also, always use hardware mirroring, duplexing or and form of RAID as compared to software mirroring, duplexing or RAID. Software mirroring, duplexing and RAID are slower than hardware mirroring, duplexing and RAID.

The speed of a server depends of various factors like the CPU, amount of memory and the access speed of a harddisk. A sluggish access of a server could be attributed to the harddisk. Check the "Dirty Cache Buffers" and "Current Disk Requests" on the Monitor console to see if the harddisk can handle the load of the access.

13. Directory Caching

Cache memory allocates memory for the hash table, the FAT, the Turbo FAT, suballocation tables, the directory cache, a temporary data storage area for files and NLM files, and available memory for other functions. The FAT and DET are written into the servers memory. The area holding directory entries is called the directory cache. The server can find a files address from the directory cache much faster than retrieving the information from disk.

Set minimum directory cache buffers to 2-3 per connection and maximum directory cache buffers to 4000. You can also set directory cache allocation wait time to 0.5 (default is 2.2). If the current Directory Cache Buffers rises above the minimum set level after the server has been up for a period of time, set the minimum Directory Cache Buffers to the current level.

NSS (Novell Storage Services), however, does not use Directory Cache Buffers. For heavy access on an NSS volume may incur some performance issues. NSS has it's own caching and parameters to be set during the loading of the NSS modules. The following are parameters which tunes the performance of NSS:

a. cachebalance
b. fileflushtimer
c. bufferflushtimer
d. closedfilecachesize
e. allocahead

"Cachebalance" controls how much memory is available for NSS to use for cache. The default for SP2 is just 10%, so increasing it to 80% or 90% puts us on a more level playing field with the legacy file system, which gets 100% (unless NSS is loaded, in which case legacy gets the remainder - 90% if NSS is using default). This percentage should be set to the percentage of total volume space taken from NSS volumes (i.e. 10GB FAT and 90GB NSS would be /cachebalance-90).

Increasing "fileflushtimer" and "bufferflushtimer" from their defaults helps the caching up to a certain point, but if increased too much causes a "throttling" problem with writing out the metadata. This is somewhat dependent on processor speed and disk speed, so you could try playing with them for your system. If NSS performance is really bad, 75 might be too high and you could try 50. If performance is just subpar, you could try increasing them a little.

"Closedfilecachesize" dictates how many closed files are kept in cache.

"Allocahead" is used to allocate extra blocks ahead of time in anticipation of new files being larger than the 4k block size, which helps performance with larger files but hurts performance with small files. Since we know most of the files will be small, we turn off allocahead.

To activate these parameters, you can issue the following command at the startup of NSS, or place it in the AUTOEXEC.NCF file:

```
  nss /cachebalance=80 /fileflushtimer=75 /bufferflushtimer=75 /closedfilecachesize=8192 /allocahead=0
```

14. Block Suballocation

Suballocation is implemented in NetWare 4.X to overcome the problem of wasted disk space due to under-allocated disk blocks (as described above). Suballocation allows multiple file endings to share a disk block. The unit of allocation within a suballocated block is a single sector (512 bytes). That means that as many as 128 file ends can occupy one 64KB block. Using suballocation, the maximum loss of data space per file is 511 bytes. This would occur when a file had one more byte than could be allocated to a full 512-byte sector. Hence, suballocation nearly eliminates the penalty of using larger disk allocation units and allows much larger disk channel transactions.

From a performance standpoint, suballocation enhances the performance of write operations within the OS by allowing the ends of multiple files to be consolidated within a single write operation. Of course, this minor improvement will often be counterbalanced by the increased overhead of managing the suballocation process. The major win is the optimization of the disk channel and cache around the 64KB disk allocation unit.

As imaging, multimedia, and other workloads involving streaming data become more prevalent, the 64KB block size will become invaluable. We recommend that everyone use the 64KB disk block size for greater efficiency, elimination of wasted space, and to take advantage of read-ahead.

It is very important to load the patches when suballocation is enabled. Suballocation does not have any SET parameters to adjust.

Everything is done automatically. It is very important to monitor the disk space to avoid suballocation problems. Novell Engineering recommends keeping 10-20 percent of the volume space free to avoid suballocation problems.

Suballocation uses free blocks to perform its function. When free blocks are low suballocation could go into "aggressive" mode, lock the volume and cause high utilization. Maintaining more than 1000 free blocks will avoid this problem in most cases. If there are not at least 1000 free blocks on the volume, run a PURGE /ALL from the root of the volume. This will free the "freeable limbo blocks" and move them back to "free blocks."

15. Disk Block Size

Based on our performance testing, we recommend a 64KB block size for all volumes. The larger 64KB allocation unit allows NetWare to use the disk channel more efficiently by reading and writing more data at once. This results in faster access to mass storage devices and improved response times for network users. If you are using RAID5 disks, set the block size equal to the stripe depth of the RAID5 disks.

16. Hotfix Blocks

Load SERVMAN, choose Storage Information, highlight the Netware Partitions in the server. Make sure that there are no "Used Hotfix blocks" shown. Also, you can "Load MONITOR", choose "Disk Information", choose the device, hit and press the key. You will see information on Redirected Blocks. "Used Hotfix blocks" and "Redirected Blocks" show that there are bad sectors on the server harddisk. You should prepare to change the harddisks of the server.

The information continues in TID2943472 (Performance, Tuning and Optimization - Part 2)

**Please select if this document solved your problem.**

| | |
|---|---|
| **Document Title:** | Performance, Tuning and Optimization - Part 1 |
| **Document ID:** | 2943356 |
| **Creation Date:** | 14OCT1998 |
| **Modified Date:** | 18MAY2000 |
| **Document Revision:** | 24 |
| **Novell Product Class:** | Connectivity Products<br>Groupware<br>NetWare<br>Novell BorderManager Services<br>Novell Directory Services |
| **Novell Product and Version:** | IP & TCPIP<br>UNIX Connectivity<br>GroupWise<br>NetWare 3.12<br>NetWare 3.2<br>NetWare 4.1<br>NetWare 4.11<br>NetWare 4.2<br>NetWare 5<br>NetWare for Small Business 4.11<br>NetWare for Small Business 4.2<br>Novell Clients<br>Novell Replication Services<br>Z.E.N.works (April 98)<br>intraNetWare 4.11<br>BorderManager 2.1 (also called 1.0)<br>BorderManager Enterprise Edition 3<br>NDS for NT 1.0 |

**DISCLAIMER**

The Origin of this information may be internal or external to Novell. Novell makes all reasonable efforts to verify this information. However, the information provided in this document is for your information only. Novell makes no explicit or implied claims to the validity of this information.

Any trademarks referenced in this document are the property of their respective owners. Consult your product manuals for complete trademark information.

# Novell Technical Information Document

**Performance, Tuning and Optimization - Part 2** *(Last modified: 17DEC1999)*

This document (**2943472**) *is provided subject to the disclaimer at the end of this document.*

**Issue**

This TID continues from TID2943356 (Performance, Tuning and Optimization - Part 1)

17. Suballocation and the Number of Free Blocks and Disk Space for Each Volume

It is also important to keep more than 1000 "free blocks" and at least 10% to 20% free disk space on each volume with suballocation enabled. Suballocation uses these free blocks to free up additional disk space. To be warned on the number of free blocks available, you can set the Volume Low Warning Threshold and the Volume Low Warning Reset Threshold to 1024.

High utilization issues can be caused by the lack of disk space. Suballocation is a low-priority thread. This means that under normal conditions, it only runs when the processor has nothing else to do and is idle. This condition of suballocation is a "non-aggressive" mode. When disk space is low, less than 10% available, suballocation can go into "aggressive" mode. Suballocation is bumped up to a regular priority thread and can take control of the volume semaphore until is has cleaned up and freed up as much space as possible. This locking of the volume semaphore causes other processes, who are trying to use the volume, to wait until the semaphore is released. In large installations, the results in an increase of Packet Receive Buffers and File Service Processes. When the Packet Receive Buffers max out, the server will start dropping connections and users are not able to login. When suballocation completes it's cleanup, the semaphore will be released and the processes on the queue will be serviced. This will result in a utilization drop and the server will return to normal operation.

The lack of free blocks is different from a lack of disk space. When files are deleted, they are kept in a "deleted" state. This means the file actually exists but is not viewable to the user and does not show up in volume statistics as used space. The number of free blocks is determined by Free Blocks = [Total Blocks - (Blocks In Use by Files + Blocks In Use by Deleted Files)]

Hence, you can have 50% of the disk available, but there are no free blocks. These blocks are use by deleted files. If free blocks are low, run a PURGE /ALL from the root of the volume to free the "freeable limbo blocks" and move them to the "free blocks" pool. To avoid doing this often, set the P (PURGE) attribute on directories that create a large amount of temporary files. The P attribute does not flow down the file system. This needs to be taken into consideration when setting the P attribute. Also, setting IMMEDIATE PURGE OF DELETED FILES = ON at the server console will avoid the purgeable files taking all the "free blocks".

18. File Compression

It is essential to have the OS patches loaded when using compression. Compression takes CPU cycles to compress and decompress files. The default set parameters for compression take this into consideration. Compression is designed to take place after midnight when most servers have little or no traffic. Also, the DAYS UNTOUCHED BEFORE COMPRESSION set parameter is designed to make sure frequently used files are not compressed. Any adjustments to the default compression SET parameters may severely impact the server's performance.

Users with disk restrictions may try to flag their home directory to IC (immediate compress) to save disk space. Flagging directories to IC will affect server performance. Normally, compression is a low priority thread, which means it only compresses files when the server is idle. When the IC flag is set, compression is bumped up to a regular priority and will not wait for idle time.

Setting DELETED FILES COMPRESSION OPTION = 2 will cause the immediate compression of files that have been deleted. This will cause high utilization because the processor is immediately compressing files upon their deletion.

Troubleshooting:
Eliminate compression as a possible problem by setting ENABLE FILE COMPRESSION=OFF. This will cause files to be queued for compression but the files will not be compressed. However, accessing compressed files causes them to be decompressed. This will eliminate compression as the cause of high utilization.

To view the amount of compression/decompression that is going on in the server, do the following:

```
  set compress screen = on
```

Hence, it is advisible to SET Days Untouched Before Compression to 30 days, and SET Minimum Compression Percentage Gain to 20.

19. File Decompression

Decompression takes up CPU cycles as well. If you are running a volume near full and compression enabled, files will be compressed and never committed decompresses due to failure of allocating enough space on the disk to hold the decompressed version.

This can be caused by "Minimum File Delete Wait Time" being set to a large value, and thus not allowing any deleted files to be reclaimed for space to commit a compressed file. The full volume situation is usually indicated by the "Compressed files are not being committed" alert on the server console. This message can be fixed by setting "Decompress Percent Disk Space Free To Allow Commit" to a number less than the current one. However, you must still remember that there must be enough space on the volume to allow for decompressed version of the file to be committed in order for that file to be committed decompressed.

As a file is decompressed, it does consume CPU cycles but it will relinquish control to allow other threads and NCPs to be serviced. A Pentium processor (60Hz) can decompress on average 1 MB a second.

20. Read After Write Verification

Read-after-write verification is an important means of protecting the data on your system. Normally, you should not disable it. However, if your disks are mirrored and reliable, you may choose to disable read-after-write verification because disabling almost doubles the speed of disk writes.

Warning:
Turning off read-after-write verification may increase the risk of data corruption on the servers hard disk. You should use the following procedure only if your disks are mirrored and reliable, and you understand the risk.

21. Garbage Collection

The GARBAGE COLLECTION process is like a disk defragmenter for the operating systems memory pool that is always running.

The purpose of the GARBAGE COLLECTION process of the operating system is to reclaim used memory nodes for a new memory scheme. Garbage collection is critical for situations where NLMs allocate small memory areas for initialization purposes then allocate larger nodes to perform the rest of its operation. Unless this smaller node memory is reallocated after the program initialization frees itself from this area, the memory will become fragmented and unusable. Therefore, the memory pool can become depleted over time.

The operating systems internal routine that handles garbage collection sorts all the nodes from the linked lists of an NLM and collapses them into larger pieces. The larger pieces are linked to the appropriate list head. If the garbage collection routine frees an entire 4 KB page, that memory is returned to cache. This internal routine can be set or interrupted and runs in the background.

Garbage Collection can be forced by a user through MONITOR. In NW4.11 under Memory Utilization, any of the system modules loaded can be selected. After it is selected, a user may press to collect garbage on that specific module or press to collect garbage on the entire system. For NW5, under Virtual Memory, Address Spaces, you can free address space memory with .

Users whose systems are running low on memory or developers who are optimizing their NLMs may need to adjust these parameters. If a NLM allocates and deallocates large blocks of memory throughout the life of the NLM, set the number of frees lower than the default. If a NLM does a lot of memory allocations and frees, set the number of frees higher than the default. If a NLM allocates and deallocates the same chunk of memory more than once, set the minimum free memory to a value higher than the size of the chunk.

Be aware that the Garbage Collection SET parameters are global; that is, they will affect all NLM garbage collection. For this reason, great care should be given in changing the default parameters. An adjustment for one NLM to improve its performance may adversely affect another NLM. Novell recommends leaving these parameters at their default unless a NLM requires a specific change.

22. Read/Write Fault Emulation and Invalid Pointers

If NetWare detects a condition that threatens the integrity of its internal data (such as an invalid parameter being passed in a function call, or certain hardware errors), it abruptly halts the active process and displays an "abend" message on the screen. ("Abend" is a computer science term signifying an ABnormal END of program.)

The primary reason for abends in NetWare is to ensure the stability and integrity of the internal operating system data. For example, if the operating system detected invalid pointers to cache buffers and yet continued to run, data would soon become unusable or corrupted. Thus an abend is NetWare's way of protecting itself and users against the unpredictable effects of data corruption.

NetWare 4 takes advantage of Intels segmentation and paging architecture. Each page of memory can be flagged present or not present, read-protected, write-protected, readable, or writable.

Exceptions caused by segmentation and paging problems are handled differently than interrupts. Normally, the contents of the program counter (EIP register) are saved when an exception or interrupt is generated. However, exceptions resulting from segmentation and paging give the operating system the opportunity to fix the page fault by restoring the contents of some of the processor registers to their state before interpretation of the instruction began. NetWare 4 provides SET parameters to enable and disable page fault emulation, giving you the choice between continuing program execution or abending.

On NW 4.x servers, the memory parameters "Allow Invalid Pointers", "Read Fault Emulation" and "Write Fault Emulation" needs to be set to OFF before analyzing a coredump or troubleshooting utilization or abending issues.

23. Interrupt Allocation and Configuration

Interrupts have priorities. The following is the order of interrupts from the highest priority to the lowest priority - 0, 1, 8, (2/9), 10, 11, 12, 13, 14, 15, 3, 4, 5, 6, 7. Interrupt 0 is used for the system timer and 1 for the Keyboard Data Ready.

For SFTIII server, you have to configure the MSL card on the highest interrupt, followed by the disk and the LAN card in that order.

"Edge triggered" interrupts may not be shared with another device but level triggered interrupts may be shared with another device. When using "level triggered" interrupts, only share the interrupt with exact types of devices, e.g. 2 NE3200.LAN or 2 AHA2940 HBA devices. Do not share interrupts with non-identical devices.

24. Set Reserved Buffers Below 16 Meg

This parameter specifies the number of file cache buffers reserved for device drivers that can?t access memory above the 16 MB. Memory address conflicts can occur when you have more than 16MB of memory in a server and the disk adapter uses 16- or 24-bit DMA or busmaster DMA. Hence, SET Reserved Buffers Below 16 Meg = 200 in STARTUP.NCF.

25. AppleTalk

There was also a problem with the ATXPR.NLM causing high utilization. The file 41MAC1.EXE contains the new version of ATXPR.NLM. The file 41MAC1.EXE is found in the NOVLIB area LIB6. Use any newer update file when found. (Current version for NW410 is MACPT3D.EXE)

26. Printing

Having a large amount of printers on a server can affect network performance. This number depends on the type of printing that is being done. Printing CAD/CAM designs take much more processor time than text documents. It is recommended, if you are concerned about utilization problems, that you set your print devices to do the processing instead of the server. This will slow down printer output but will relieve the utilization on the file server.

Reference Literature:

Novell Research:
May 1993 - NetWare 4.0 Performance Tuning and Optimization: Part 1
June 1993 - NetWare 4.0 Performance Tuning and Optimization: Part 2
October 1993 - NetWare 4.x Performance Tuning and Optimization: Part 3
February 1995 - Resolving Critical Server Issues
March 1995 - Server Memory
April 1995 - MONITOR.NLM
June 1995 - Abend Recovery Techniques for NetWare 3 and 4 Server
November 1995 - Server Memory
March 1997 - IntranetWare Server Automated Abend Recovery
March 1997 - Optimizing IntranetWare Server Memory

TIDs:
TID14270 - NetWare 4.0 Memory Allocation - Part 1 of 2
TID14271 - NetWare 4.0 Memory Allocation - Part 2 of 2
TID1005436 - High Utilization and Suballocation
TID1005736 - Compression and High Utilization
TID1007561 - NetWare OS Patches
TID1005963 - Troubleshooting NetWare 4.1 High Utilization
TID1202046 - NetWare 3.x and 4.x Directory Entry Limits
TID2905856 - Additional Notes for High Utilization
TID2906943 - PC INTERRUPTS
TID2917538 - Suggestions for Troubleshooting Abends
TID2923372 - SFTIII configuration
TID2950154 - 8259 Interrupts and NetWare

Novell Press:
Novell's Guide to Resolving Critical Server Issues by Rich Jensen and Brad Dayley

Others:
High Utilization by Rich Jardine, Product Support Engineer, NetWare OS Support, Novell Inc. (28 Sep 95)
Suballocation by Rich Jardine, Product Support Engineer, NetWare OS Support, Novell Inc. (1 Feb 96)

Compression by Kyle Unice, Novell Software Engineer (1 Feb 96)
FYI.P.12444 - What Does the Garbage Collection SET Parameter Do?
November 1993 - NetNotes

File Downloads:
HIGHUTL1.EXE - Troubleshooting High Utilization for NW4
TABND2A.EXE - Diagnostic programs / utilities to troubleshoot Abends
CONFG8B.EXE - CONFIG.NLM and CONFGNUT.NLM to collect server configuration
CFGRD6B.EXE - The Config Reader, Internet Edition! (version 2.67)

Note: Kindly access the Novell Support Connection web site (http://support.novell.com or http://support.novell.com.au) for all the latest patches, fixes and file updates. You will also find a knowledge base which accesses the latest Novell Technical Information Database (TID) to help you be in-touch with the latest issues so that you can pro-actively manage your servers and the network.

**Please select if this document solved your problem.**

| | |
|---|---|
| **Document Title:** | Performance, Tuning and Optimization - Part 2 |
| **Document ID:** | 2943472 |
| **Creation Date:** | 16OCT1998 |
| **Modified Date:** | 17DEC1999 |
| **Document Revision:** | 10 |
| **Novell Product Class:** | Groupware<br>Management Products<br>NetWare<br>Novell Directory Services |
| **Novell Product and Version:** | GroupWise<br>ManageWise<br>NetWare 3.11<br>NetWare 3.12<br>NetWare 3.2<br>NetWare 4.1<br>NetWare 4.11<br>NetWare 4.2<br>NetWare 5<br>NetWare for Small Business 4.2<br>Novell Clients<br>Novell Replication Services<br>Z.E.N.works (April 98)<br>intraNetWare 4.11<br>NDS for NT 1.0 |

## DISCLAIMER

The Origin of this information may be internal or external to Novell. Novell makes all reasonable efforts to verify this information. However, the information provided in this document is for your information only. Novell makes no explicit or implied claims to the validity of this information.

Any trademarks referenced in this document are the property of their respective owners. Consult your product manuals for complete trademark information.