IBM Tivoli Storage Manager

Performance Tuning Guide

SC32-9101-03

Eighth Edition (October 2005)

Americas Advanced Technical Support

This edition is intended as supplementary documentation for the currently supported levels of IBM Tivoli Storage Manager Servers and Clients and is current until superseded by technical newsletters or another edition of this document. For the latest tuning information, be sure you are using the latest edition of this document. Be sure to search for the latest tuning advice on IBM's software support site.

Note
Before using this information and the product it supports, be sure to read the general information under "Notices".

We are very interested in hearing about your experience with Tivoli products and documentation. We also welcome your suggestions for improvements. If you have comments or suggestions about our documentation, please complete our customer feedback survey at www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html by selecting the Feedback link in the left navigation bar.

If you have questions or comments regarding Tivoli publications and product documentation, please visit www.ibm.com/software/tivoli/contact.html to send an e-mail.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2005. All rights reserved.
Note to U.S. government Users -- Documentation related to restricted rights -- Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule contract with IBM Corp.

Table of Contents

IBM Tivoli Storage Manager 1
Performance Tuning Guide 1
SC32-9101-03 1
Table of Contents 2
About this Publication 4
Notices 4
  Disclaimer 4
  Programming Interface Information 4
  Trademarks 4
Summary of Changes for this Edition 5
Chapter 1. General Overview of TSM Tuning 5
  Overview of TSM Performance 5
Chapter 2. TSM Tuning Options 7
  TSM Server Tuning Considerations 7
    Tuning Options 7
    BUFPoolsize 7
    EXPInterval 8
    LOGPoolsize 9
    MAXNUMMP 9
    MAXSessions 10
    MOVEBatchsize / MOVESizethresh 10
    RESTOREINTERVAL 11
    SELFTUNEBUFPOOLsize 11
    SELFTUNETXNsize 12
    TAPEIOBUFS 12
    TCPNodelay 13
    TCPWindowsize 13
    TXNGroupmax 14
    USELARGEbuffer 15
  Platform Specific Recommendations 16
    OS/390 Server 16
    OS/400 PASE Server 18
    MS Windows Server 19
    Solaris Server 22
    AIX Server 23
    Linux Server 25
    HP-UX Server 26
  TSM Client Tuning 28
    Tuning Options 28
    COMPRESSION 28
    COMPRESSALWAYS 29

COMMRESTARTDURATION / COMMRESTARTINTERVAL 30
QUIET 30
DISKBUFFSIZE 31
LARGECOMmbuffers 31
MEMORYEFFICIENTBACKUP (previously SLOWINCREMENTAL) 32
Multi-Session Backup/Restore 32
RESOURCEUTILIZATION 33
Table 2. Resource Utilization Settings 34
TAPEPrompt 36
TCPBufsize 36
TCPNodelay 36
TCPWindowsize 36
TXNBytelimit 37
General Performance Considerations 38
Platform Specific Recommendations 38
AIX Client 38
Linux Client 39
Windows Client 39
Apple Macintosh OS Client 40
DEC Client 41
Novell Client 41
SOLARIS Client 42
Command Line Options 43
Chapter 3. TSM Optimization 44
Server Optimization 44
LAN Free Backup 47
TSM Storage Agent Tuning 48
TDP for Domino for S/390 48
External TSM Tuning Tips 48
Chapter 4. Storage Device Hints and Tips 53
IBM LTO Drives 53
SSA Disks 55
SilverNode SCSI Slot Positioning 56
Busses 57
Appendix A. Using TSM's Archive Function 57
Abstract 57
History 57
Implementing Archive 59
Symptoms 59
Suggested Actions 60
Summary 61
Appendix B. Network Protocol Tuning 61
TCP/IP Tuning Considerations 61
Communication Concepts 61
Platform Specific Recommendations 64
Glossary 78

References 82
Other Resources 82
IBM LTO Related Publications 82
Additional TCP/IP Performance Related Publications 82
Referenced Web Sites 83

About this Publication

This publication is intended to help you in performance tuning IBM Tivoli Storage Manager. This document should be used in conjunction with the IBM Tivoli Storage Manager product publications: *Administrator's Guide, Administrator's Reference* and the client *User's Guides.*

Notices

Disclaimer

The information contained in this document is distributed on an "as is" basis ***without any warranty either expressed or implied.*** While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere.

Measured performance data contained in this document was determined in a controlled environment, and therefore, the results which may be obtained in other operating environments may vary significantly.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM licensed program in this document is not intended to state or imply that only IBM's program maybe used. Any functionally equivalent program may be used instead.

Programming Interface Information

This publication is intended to help you use the Tivoli Storage Manager.

This publication documents information **not** intended to be used as programming interfaces of IBM Tivoli Storage Manager.

Trademarks

IBM, Tivoli, AIX, MVS/ESA, MVS/XA, OS/2, OS/390, z/OS, OS/400, VM/ESA, VSE/ESA, eServer, pSeries, zSeries, xSeries, iSeries, S/390, RISC System/6000, RS/6000, SP, PowerPC, AS/400, VTAM, C/370, Database 2, DB2, DB2 Universal Database, and RACF are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Intel and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java or all Java-based trademarks and logos, and Solaris are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Summary of Changes for this Edition

TSM Client Tuning: Added section for DISKBUFFSIZE. Changed default values for TCP/IP and transaction settings.

TSM Server Tuning: Added warning about simultaneous disk storage pool volume format commands

AIX Server Platform Specific Recommendations: Removed incorrect warning regarding use of raw volumes in HACMP environments.

Linux Server Platform Specific Recommendations: Removed reference to Raw Logical Volumes. Raw Logical Volumes are not supported on Linux

Apple Macintosh OS Client Platform Specific Recommendations: Added this section.

External Tuning – Network Recommendations: Added recommendation for sizing AIX minfree and maxfree

Chapter 1. General Overview of TSM Tuning

Overview of TSM Performance

TSM performance can be influenced by various tuning parameters. Tuning these functions for good performance requires diligence and expertise on the part of the implementer. Tuning TSM can be quite complex because of the many operating systems, network configurations and storage devices that TSM will support. It is the purpose of this tuning guide to provide tips and hints to ease the task of getting good performance for TSM operations.

Performance tuning for a single platform function is quite complex, but due to years of experience, it is generally well understood. TSM, however, functions in the client/server realm, supports many operating systems, works across networks, and accepts different communication protocols. Consequently, there are many more factors that affect performance. These factors can affect TSM performance significantly:

- Average client file size
- Percentage of files changed since last incremental backup
- Percentage of bytes changed since last incremental backup
- Client hardware (CPUs, RAM, disk drives, network adapters)
- Client operating system
- Client activity (non-Tivoli Storage Manager workload)
- Server hardware (CPUs, RAM, disk drives, network adapters)
- Server storage pool devices (disk, tape, optical)
- Server operating system
- Server activity (non-Tivoli Storage Manager workload)
- Network hardware and configuration
- Network utilization
- Network reliability
- Communication protocol
- Communication protocol tuning
- Final output repository type (disk, tape, optical)

Clearly, with this many combinations, it is not feasible to discuss all possible combinations of these parameters within the scope of this guide. The topics will be limited to those that you can control somewhat without replacing hardware.

This guide is divided into four major sections:

1. General Tuning Overview
2. TSM Tuning Options
3. TSM Optimization
4. Storage Device Hints and Tips

Each section provides a brief overview of the four topics, followed by details of the various parameters affecting performance.

Defaults and recommended values for good performance are provided for each parameter, where appropriate. There are many terms used throughout the following sections, mostly

unique to either TCP/IP, NetBIOS or TSM. In the appendix are two glossaries, one for NetBIOS, and one for TCP/IP.

Chapter 2. TSM Tuning Options

TSM Server Tuning Considerations

This chapter focuses on tuning the TSM server and client to obtain maximum performance. It is a discussion of the tuning parameters applicable to all TSM servers and clients, followed by server and client platform specific recommendations.

Tuning Options

The following parameters are tunable on most TSM servers. Not all options are supported on all server platforms. Check the *Administrators Reference* for applicability to a particular platform. These parameters can all be changed in server options file(dsmserv.opt) or some can be changed with the server SETOPT command. For any changes made in the server options file, the server must be halted and then restarted for the changes to take effect.

- BUFPoolsize
- EXPInterval
- LOGPoolsize
- MAXNUMMP (Change in the client NODE definition)
- MAXSessions
- MOVEBatchsize / MOVESizethresh
- RESTOREINTERVAL
- SELFTUNEBUFPOOLsize
- SELFTUNETXNsize
- TAPEIOBUFS
- TCPNodelay
- TCPWindowsize
- TXNGroupmax
- USELARGEbuffer

BUFPoolsize

The database buffer pool provides cache storage, which allows database pages to remain in memory for longer periods of time. When database pages remain in cache, the server can make continuous updates to the pages without requiring I/O operations to external storage. While a larger database buffer pool can improve server performance, it will also require more memory.

An optimal setting for the database buffer pool is one in which the cache hit percentage is greater than or equal to 99%. Performance decreases drastically if the buffer pool cache

hit ratio decreases below 99%. To check the cache hit percentage, use the **query db format=detail.** Increasing the BUFPoolsize parameter can improve the performance of many TSM server functions such as multi-client backup, storage pool migration, storage pool backup, expiration processing and move data. If the cache hit percentage is lower than 99%, increase the size of the BUFPoolsize parameter in the DSMSERV.OPT file (It is necessary to shutdown and restart the server, for any change to take effect) or use the SETOPT command.

For most servers, we recommend starting with a value of 32768, which equals 8192 database pages. Based on your cache hit rate, increase in 1MB increments. A cache hit percentage greater than 99% is an indication that the proper BUFPoolsize has been reached. However, continuing to raise BUFPoolsize beyond that level can be very helpful. While increasing BUFPoolsize, care must be taken not to cause paging in the virtual memory system. Monitor system memory usage to check for any increased paging after the BUFPoolsize change. (Use the RESET BUFP command to reset the cache hit statistics.) If you are paging, buffer pool cache hit statistics will be misleading since the database pages will be coming from the paging file.

We have found the optimal size to be 1/8 to 1/2 of real memory.

For AIX, to avoid paging, see the section on 'AIX's VMTUNE'.

See SELFTUNEBUFPOOL regarding automatic tuning of the BUFPOOLSIZE

TABLE 1. BUFPOOLSIZE Recommended Starting Point

Recommendations for AIX, HP, Sun, Windows Servers		
System Memory (MB)	**Recommended Buffer Pool Size (MB)**	**Recommended Buffer Pool Size (KB)**
32	2	2048
48	3	3072
64	4	4096
96	9	9216
128	14	14336
160	20	20480
256	32	32768
512	64	65536
1024	128	131072
2048	256	262144

EXPInterval

Specifies the interval, in hours, between an automatic inventory expiration run by the TSM server. Inventory expiration removes client backup and archive file copies from the server.

Backup and archive copy groups can specify the criteria that make copies of files eligible for deletion from data storage. However, even when a file becomes eligible for deletion, the file is not deleted until expiration processing occurs. If expiration processing does not occur periodically, storage pool space is not reclaimed from expired client files, and the TSM server requires increased disk storage space.

Expiration processing is very CPU and I/O intensive. If possible, it should be run when other TSM processes are not occurring. To enable this, either schedule expiration once per day, or set EXPInterval to 0 and manually start the process with the EXPIRE INVENTORY command at the server. The default is 24. Expiration processing can also be scheduled in an administrative schedule.

When using the DURATION parameter through an administrative schedule, periodically check that expiration is actually completing within the specified time.

Recommendation
EXPInterval 0 (which means no expiration processing) and use an administrative schedule to execute expiration at an appropriate time each day.

LOGPoolsize

Specifies the size of the recovery log buffer pool size in kilobytes. A large recovery log buffer pool may increase the rate by which recovery log transactions are committed to the database, but it also requires more memory. The recovery log buffer pool is used to hold new transaction records until they can be written to the recovery log. The size of the recovery log buffer pool can affect the frequency in which the server forces records to the recovery log. To determine if the LOGPoolsize should be increased, monitor the value of LogPool Percentage Wait. To check the wait percentage, use the **query log format=detail.** If the value is greater than zero, increase the value of LOGPoolsize. As the size of the recovery log buffer pool is increased, remember to monitor system memory usage.

Recommendation
LOGPoolsize 2048 - 8192

MAXNUMMP

The MAXNUMMP setting for the node, in the server, specifies the maximum number of mount points a node is allowed to use on the server and can be set to an integer from 0-

999. Zero specifies that the node cannot acquire any mount point for a backup or archive operation. However, the server will still allow the node a mount point for a restore or retrieve operation. If the client stores its data to a storage pool that has copy storage pools defined for simultaneous backup, the client may require additional mount points. As a general rule, you must assign one mount point for each copy storage pool of sequential device type. If the primary storage pool is of sequential device type, then assign a mount point for the primary storage pool as well.

MAXSessions

Specifies the maximum number of simultaneous client sessions that can connect with the TSM server. The default value is 25 client sessions. The minimum value is 2 client sessions. The maximum value is limited only by available virtual memory or communication resources. This parameter specifies the maximum number of simultaneous client sessions that can connect with the TSM Server. By limiting the number of clients, server performance can be improved, but the availability of TSM services to the clients is reduced. Set this parameter in the TSM server options file (TSM must be halted and then restarted) or use the SETOPT command.

Recommendation
MAXSessions 25 (default)

MOVEBatchsize / MOVESizethresh

This option specifies the number of files that are to be moved and grouped together in a batch, within the same server transaction. The default value for MOVEBatchsize is 40 and the maximum value is 1000. The default value for MOVESizethresh is 500 and the maximum value is 2048.

MOVEBatchsize and MOVESizethresh options help tune the performance of server processes that involve the movement of data between storage media. These processes include storage pool backup and restore, migration, reclamation, and move data. These options are specified in the server options file.

The number of client files moved for each server database transaction during a server storage pool backup/restore, migration, reclamation or move data operation will be determined by the number and size of the files in the batch. If the number of files in the batch equals the MOVEBatchsize before the cumulative size of the files becomes greater than the MOVESizethresh, then the MOVEBatchsize is used to determine the number of files moved or copied in the transaction. If the cumulative size of files being gathered for a move or copy operation exceeds the MOVESizethresh value before the number of files becomes equivalent to the MOVEBatchsize, then the MOVESizethreshvalue is used to determine the number of files moved or copied in the transaction.

Note that when the MOVEBatchsize or MOVESizethresh parameters are increased from their default values, the server will require more space in the recovery log. The recovery log may require an allocation of space two or more times larger than a recovery log size which uses the defaults. In addition, the server requires a longer initialization time at startup. The impact of a larger Recovery Log size will be felt while running the server with the logmode set to NORMAL (the default value). If you choose to increase these values for performance reasons, be sure to monitor recovery log usage during the first few storage pool backup/restore, migration, reclamation, or move data executions to ensure sufficient recovery log space is available.

The DEFINE LOGVOLUME and EXTEND LOG commands are used to add space to the server recovery log. Also, ensure additional volumes are available (formatted with the DSMFMT or ANRFMT program) for extending the recovery log.

Recommendation
MOVEBatchsize 1000 MOVESizethresh 2048

RESTOREINTERVAL

Restartable restores allow restores to continue after an interruption without starting at the beginning. This reduces duplicate effort or manual determination of where a restore process was terminated. RESTOREINTERVAL defines the amount of time an interrupted restore can remain in the restartable state.

Specifies how long, in minutes, that a restartable restore session can be in the database before it can be expired. The minimum value is 0. The maximum is 10080 (one week). The default is 1440 minutes (24 hours). If the value is set to 0 and the restore is interrupted or fails, the restore is still put in the restartable state. However, it is immediately eligible to be expired. Restartable restore sessions consume resources on the TSM server. You should not keep these sessions any longer than they are needed.

Recommendation
RESTOREINTERVAL (Tune to your environment)

SELFTUNEBUFPOOLsize

Controls the automatic adjustment of the buffer pool size and is set to Yes or NO. The default is NO on the server. Specifying YES causes the data base cache hit ratio statistics to be reset prior to starting expiration processing and examined AFTER expiration processing completes. The buffer pool size is adjusted if the cache hit ratio is less than 98%. The percentage increase in buffer pool size is half the difference between the 98% target and the actual buffer pool cache hit ratio. This increase is not done if a platform-specific check fails:

UNIX
: buffer pool size may not exceed 10% of physical memory.

Windows
: same as UNIX with an additional check for memory load not exceeding 80%.

MVS
: may not exceed 50% of region size.

Be careful when using SELFTUNEFBUFPOOLSIZE on Unix and Windows systems. Since the upper limit is small (10% or real memory), the self-tuning algorithm might not achieve the optimal size. Monitor the buffer pool hit ratio and tune manually if necessary.

You must have sufficient physical memory. Lack of sufficient physical memory means that increased paging will occur. The self-tuning process gradually raises the buffer pool size to an appropriate value. You should use TABLE 1 to set the initial size (with the BUFPOOLSIZE option) and to ensure that the buffer pool size is appropriate each time the server starts.

Recommendation
SELFTUNEBUFpoolsize Yes

SELFTUNETXNsize

Specifies whether TSM can automatically change the values of the TXNGROUPMAX, MOVEBATCHSIZE, and MOVESIZETHRESH server options. TSM sets the TXNGROUPMAX option to optimize client-server throughput and sets the MOVEBATCHSIZE and MOVESIZETHRESH options to their maximum to optimize server throughput. The default is NO.

This parameter affects server data movement functions.

Note: Larger transaction sizes will consume more space in the TSM recovery log when running in Normal mode. To prevent running out of log space when in Roll Forward mode, we recommend using the **DEFINE SPACETRIGGER** command to create a log space trigger

Recommendation
SELFTUNETXNsize Yes

TAPEIOBUFS

To use the BSAM overlap I/O buffering methods, a new server option is available:

TAPEIOBUFS number of buffers

z/OS Platform Only

The number of buffers specifies a number from 1 to 9. If 1 is specified, then Tivoli Storage Manager does not use the overlapping I/O for 3590 tape media. If a number greater than 1 is specified, Tivoli Storage Manager allocates the buffers based on the formula shown above.

The use of this option may increase the I/O throughput with the 3590 tape media, but it requires more memory allocation for the address space. An optimized throughput scenario is using this option set to 9, with the UNIX System Service Socket and a Gigabit Ethernet.

Recommendation
TAPEIOBUFS 1 (default)

Note: This option does not apply to device classes of the 3590 device type on servers running OS/390 Version 2 Release 10 or higher and on z/OS.

TCPNodelay

Specifies whether the server allows data packets that are less than the maximum transmission unit (MTU) size to be sent out immediately over the network.

When set to NO, the server will buffer data packets that are less than the MTU size:

- Buffering can improve network utilization
- Buffering requires a delay that can impact session throughput greatly

When set to Yes, it will disable the TCP/IP Nagle algorithm which allows data packets less than the MTU size to be sent out immediately. Setting this option to YES might improve performance in higher speed networks. The default is Yes.

Recommendation
TCPNodelay Yes

Note: This option is also valid on the TSM client.

TCPWindowsize

The TCPWindowsize option specifies the amount of receive data in Kilobytes that can be in-transit at one time on a TCP/IP connection. The sending host cannot send more data

until an acknowledgement and TCP receive window update are received. Each TCP packet contains the advertised TCP receive window on the connection.

A larger window allows the sender to continue sending data, and may improve communication performance, especially on fast networks with high latency. It is important to recognize that to improve backup performance, increase the TCP receive window on the TSM server. To improve restore performance, increase the TCP receive window on the client.

The TCPWindowsize option overrides the operating system's TCP send and receive spaces. In AIX, for instance, these parameters are tcp_sendspace and tcp_recvspace and are set as “no” options. For TSM, the default is 63KB, and the maximum is 2048KB. Specifying TCPWindowsize 0, will cause TSM to use the operating system default. This is not recommended because TSM’s optimal setting might not be same as the optimal setting for other applications.

The TCPWindowsize option specifies the size of the TCP sliding window for all clients and all servers. A larger window size can improve communication performance, but uses more memory. It enables multiple frames to be sent before an acknowledgment is obtained from the receiver. If long transmission delays are being observed, increasing the TCPWindowsize may improve throughput.

In AIX, rfc1323 must be set on to have window sizes larger than 64K-1,

Recommendation #1
TCPWindowsize 63
Recommendation #2 (SP2 environment with a high speed switch)
TCPWindowsize 128 - 640
Reommendation #3 (Gigabit Ethernet with Jumbo Frames – 9000 MTU)
TCPWindowsize 128

Note: This option is also valid on the TSM client. Use 128 for Jumbo Frame Gigibit Ethernet.

TXNGroupmax

Specifies the number of files transferred as a group between commit points. This parameter is used in conjunction with the TXNBytelimit client option. The default is 256. The maximum is 65000. You can override the value of this option for individual client nodes with the TXNGROUPMAX parameter in the REGISTER NODE and UPDATE NODE commands.

The TXNGroupmax option is used in conjunction with the client TXNBytelimit option (Refer to "TXNBytelimit"). This option reduces the number of server transactions by

increasing the number of files within any one transaction. Therefore, the amount of overhead during backup or restore caused by database commits is reduced.

- Increasing the number of files per transaction will increase recovery log requirements on the server. Check log and log pool space to ensure there is enough space.
- Increasing the number of files per transaction may result in more data being retransmitted if a retry occurs. This may negatively affect performance.

The benefits of changing this parameter are dependent on configuration and workload characteristics. In particular, this parameter benefits tape storage pool backup more so than disk storage pool backup, especially if many small files are in the workload.

When setting the size of transactions, consider setting a smaller size if you are suffering many resends due to compression or files changing during backup when using static, shared static, or shared dynamic. This would apply to static as well as to shared because when the client realizes a file has changed during a backup and decides to not send it, the file that is, it would still have to re-send the other files in that transaction.

Lab tests have shown that settings higher than 4096 give no benefit. Set TXNGroup to 256 in your server options file. If some clients have small files and go straight to a tape storage pool then consider raising TXNGroupmax to higher values (up to 4096) for those clients only.

Recommendation
TXNGroupmax 256 in server options files, higher values (max 4096) for only those clients with small files and direct to tape backup

USELARGEbuffer

The USELARGEbuffer server option is now the default setting. This option increases communication and device I/O buffers. Both the client/server communication buffer and disk device I/O buffers are increased from 32 KB to 256 KB when this option is set to YES. The communication I/O buffer is used during data transfer with a client session, such as a backup session. The disk I/O buffer is used when data is read from or written to a disk storage pool.

Optimal AIX TSM server performance in the lab, with USELARGEbuffer YES was achieved by increasing the amount of AIX disk read-ahead. See "AIX'sVMTUNE".

Significant improvement in data transfer operations and CPU usage has been observed when the USELARGEbuffer feature is enabled. Increasing the buffer sizes allows client/server communications and disk I/O to be more efficiently performed than in

environments without this feature enabled. Reads and writes are quicker and server resources are better utilized. By reducing CPU utilization, more clients can concurrently be serviced, improving overall system performance. These benefits are complemented with server aggregation. Aggregation groups smaller, logical client files into fewer but larger physical files at the server level. Larger files are better able to take advantage of the larger buffers.

Recommendation
USELARGEbuffer Yes

Platform Specific Recommendations

OS/390 Server

Performance Recommendations

The following recommendations may help you provide optimized Tivoli Storage Manager performance for your environment.

- Place the TSM server database and storage pool volumes on the fastest I/O disk such as ESS. Also provide enough separate physical disks to minimize I/O contention.
- A region size of 256MB (REGION=256M) is recommended for starting the Tivoli Storage Manager server. However, this value can be decreased or increased based on server workload. To determine the optimum value for your installation, monitor server operation and set the region size according to the results.

  If the specified region is too small, server performance can be significantly impacted, especially during periods of high server activity. For example, the operating system GETMAIN and FREEMAIN processing can have a major impact on the performance of transaction-oriented applications such as Tivoli Storage Manager. To eliminate or minimize GETMAIN and FREEMAIN calls, the server uses its own method for satisfying storage requests. The ability of the server to avoid calls to the GETMAIN and FREEMAIN procedures is highly dependent on adequate region size for the workload.

  When increasing the region size, use 128MB increments until the desired performance is achieved, or until it is clear that the additional storage is not yielding improved performance. Once this occurs, you can reduce the region size in small increments (for example, 32 MB), to determine the optimum region size for your workload. It is important that performance be monitored for a period of time and over periods of high activity.

  A region size of 0M (REGION=0M) is not recommended. Specifying 0M will result in poor server performance.

- VSAM I/O pages can be fixed to allow a faster throughput of data for operations involving the database, the recovery log, and all storage pool volumes. Specifying a size for the VSAM pages can significantly reduce the locking and unlocking of VSAM pages while operations are occurring to and from disk storage pool volumes.
- Use TSM server option txngroupmax 256 and TSM client option txnbytelimit 25600 to maximize the transaction size. A larger transaction size increases the size of TSM server file aggregates. File aggregation provides throughput improvements for many server data movement and inventory functions, such as storage pool migration, storage pool backup, and inventory expiration. A larger transaction size when using backup directly to tape reduces the number of tape buffer flushes and therefore can improve throughput.
- Use TSM server option uselargebuffers yes to improve throughput and server efficiency for large files.
- When enablelanfree is set to no, use TSM client option largecommbuffers yes for optimal performance with largecommbuffers on an AIX TSM client. Tune JFS for good disk read-ahead for large files using vmtune -R256 -f120 -F376. Note: largecommbuffers has been replaced with diskbuffsize for TSM 5.3 and later clients.
- The client option Resource Utilization can be set to value other than default if the backup on the client involves many files and they are spanning or residing on multiple physical disks. A value of 5 or greater is recommended for the Resource Utilization option (RESOURCEUTIL nn) in client options file. However, for optimal utilization of the TSM environment, you need to evaluate the load of TSM server, network bandwidth, client CPU and I/O configuration and take into consideration before changing Resource Utilization option.
- For optimal TSM backup and restore performance when using fast clients and heavily loaded network or server, use TSM client compression. For optimal TSM backup and restore performance when using a slow client, or a lightly loaded network or server, do not use TSM compression. However, be sure to consider the trade-off of greater storage requirements on the server when not using client compression.
- If the z/OS or OS/390 processor is connected to the LAN using OSA/E GbE adapter, use the following TCP/IP settings: TCP/IP (CS OS/390) Profile dataset: Gateway stanza; Packet-size 9000 TCPCONFIG stanza; TCPSENDBFRSIZE 262144 and TCPRCVBFRSIZE 262144 - AIX TSM client option; TCPWindowsize 256 - AIX Gigabit Ethernet; Transmit and Receive jumbo frames YES Software transmit queue size 1024 Maximum IP Packet Size for this device 9000 - AIX no options; rfc1323=1 sb_max=1048578 thewall=524340
- If the z/OS or OS/390 processor is connected to the LAN using OSA/E FENET adapter, use the following TCP/IP settings: TCP/IP (CS OS/390) Profile dataset: Gateway stanza; Packet-size 1500 TCPCONFIG stanza; TCPSENDBFRSIZE 65535 and TCPRCVBFRSIZE 65535 - AIX TSM client option; TCPWindowsize 64 - AIX no options: rfc1323=1 sb_max=131072 thewall=131072
- TCP/IP profile dataset, GATEWAY statement: Use 9000 byte packet size for jumbo Ethernet frames or use 1500 byte packet size for standard Ethernet frames

The TSM OS/390 server now has the capability to exploit multiple CP's. This option is enabled by putting MPTHREADING YES in the server option file. At this time, the default for MPTHREADING is OFF. We believe that only customers that are forced to run multiple concurrent server processes (reclamation, migration, etc.) should use this option. And then, should carefully monitor CPU usage closely.

Server Options File
Recommended Options
MPTHreading Yes
USELARGEBuffer Yes
BUFPoolsize 32768
LOGPoolsize 1024
TXNGroupmax 256
MOVEBatchsize 1000
MOVESizethresh 2048

OS/400 PASE Server

- Formatting disk storagepool volumes which will be placed in the same filesystem with parallel running dsmfmt processes can cause the storagepool volumes to be highly fragmented. For disk storage pool volumes in the same filesystem, do not issue multiple dsmfmt or DEFINE VOLUME commands simultaneously. Format disk storagepool volumes sequentially, one at a time, if they are placed into the same filesystem. This should create files which have only a few gaps and will improve sequential read/write performance.

dsmserv.opt
Recommended Options
BUFPoolsize (see Table 1)
LOGPoolsize 512
TXNGroupmax 256
MOVEBatchsize 1000
MOVESizethresh 2048
TCPWindowsize 63
USELARGEBuffer Yes
TCPNODELAY Yes

MS Windows Server

Performance Recommendations

The Tivoli Storage Manager server for Windows has been optimized for performance with several internal code changes. The biggest performance improvements occur with multiple active server threads on multi-processor (SMP) machines. These changes are transparent to the user and require no additional tuning. A major performance enhancement of Tivoli Storage Manager is the ability to use multiple sessions with the server during backup. This is accomplished using the client RESOURceutilization number option, where number specifies the level of resources the TSM server and client can use during processing.

Using the NTFS filesystem on the disks used by the TSM server will most often be the best choice. These disks include the TSM recovery log, TSM database, and TSM storage pool volumes. Although there is generally no performance difference for TSM functions between using NTFS and FAT on these disks, NTFS has the following advantages:

- Support for larger disk partitions
- Better data recovery
- Better file security
- Formatting TSM storage pool volumes on NTFS is much faster

Keep in mind that using NTFS on the TSM client does cause a performance degradation for small file workloads relative to FAT, because of the need to check security information during backup and the impact of the NTFS filesystem recovery capability during restore.

- NTFS file compression should not be used on disk volumes that are used by the TSM server, because of the potential for performance degradation.

- For optimal Tivoli Storage Manager for Windows server performance with respect to Windows real memory usage, use the server property setting for Maximize Throughput for Network Applications. This setting gives priority to application requests for memory over requests from the Cache Manager for file system cache. This setting will make the most difference in performance on systems that are memory constrained.

- For optimal Tivoli Storage Manager backup and restore performance with a large number of clients, consider using Tivoli Storage Manager client compression. Compressing the data on the client reduces demand on the network and on the Tivoli Storage Manager server. The reduced amount of data on the server continues to provide performance benefits whenever this data is moved, such as for storage pool migration and storage pool backup. However, client compression may significantly reduce the performance of each individual client, and the reduction will be more pronounced on the slowest client systems.

Optimizing Windows NT DPC Distribution

When using a multiprocessor server for the Tivoli Storage Manager server, optimal DPC distribution among those processors may improve backup performance. A deferred process call (DPC) results from an interrupt from a network adapter when a data packet arrives. Some multiprocessor servers can dynamically distribute adapter interrupts to the least busy processor. This hardware feature is helpful in preventing processor bottlenecks and poor network performance in systems that service many networking requests. This feature is often referred to as symmetric interrupt distribution and is designed to improve scaling and to prevent a single processor from becoming a bottleneck while other processors have excess capacity. It is available on the Windows NT 4.0 Hardware Abstraction Layer (HAL) for the Pentium family of processors and is also supported on Windows 2000. By default, Windows NT 4.0 does not take advantage of symmetric interrupt distribution and assigns DPC activity associated with a network adapter to an associated processor. Each network adapter is associated with a single processor, begining with the highest numbered processor. If a processor frequently operates at capacity (Performance Monitor Processor: % Processor Time > 95%) and more than half of its time is spent servicing DPCs (Processor: % DPC Time > 50%), it is possible to improve performance by balancing this DPC activity across processors.

This is accomplished by adjusting the ProcessorAffinityMask registry entry, located in:

HKEY_LOCAL_MACHINE System CurrentControlSet Services NDIS Parameters

Warning! Editing the registry incorrectly can cause serious problems that may require reinstallation of the operating system. It is recommended that you back up the contents of the registry prior to performing modifications so that the contents can be restored in case of problems. The default value of ProcessorAffinityMask is x'ffffffff'. To balance the network DPC activity across multiple processors, set ProcessorAffinityMask to x'0' and reboot the system.

Use the TSM server/client option **TCPwindowsize** to improve network throughput during backup/restore and archive/retrieve operations.

The maximum TCP receive window supported under Windows NT 3.5, 3.51, and 4.0 is 65535 (64K - 1) bytes. Using an TSM ***tcpwindowsize*** value that results in a larger window size being requested actually results in the default Windows NT receive window,

which may be substantially smaller than what is expected. We therefore recommend not exceeding a value of 63 for the TSM ***tcpwindowsize*** option in these environments. Windows 2000/XP does provide a larger TCP receive window size when communicating with hosts that also provide this support, known as RFC1323. In these environments, a TSM ***tcpwindowsize*** greater than 63 may be useful with Windows 2000/XP systems that are communicating exclusively with other Windows 2000/XP or UNIX systems.

A window size larger than the amount of buffer available on the network adapter may cause throughput to degrade by 10X or more due to the need to constantly resend packets that were lost on the adapter. Most adapters have much more memory available now, however, the point is that the downside risk is very great for exceeding the hardware capabilities, but the upside performance improvements are more likely limited to 20% to at best 50%. Lab tests show no real measurable performance improvement , partly because Ethernet is a low latency network. TR and FDDI are higher latency (because of the need to wait for access to the token), so they may show more improvement. Anything talking to MVS seems to benefit with increased tcpwindowsize.

For optimal Tivoli Storage Manager backup and restore performance when using a local client, use the named pipes communication method. For Windows in the Network Properties box, move the NetBEUI protocol to first in the list of protocol bindings for the Workstation service.

Miscellaneous TSM Client/Server Issues

- Anti-virus software can negatively impact backup performance
- Disable or do not install unused services
- Disable or do not install unused network protocols
- Favor background application performance
- No screen savers
- Make sure the paging file isn't fragmented
- Keep device drivers updated, especially for new hardware

Formatting disk storagepool volumes which will be placed in the same filesystem with parallel running dsmfmt processes can cause the storagepool volumes to be highly fragmented. For disk storage pool volumes in the same filesystem, do not issue multiple dsmfmt or DEFINE VOLUME commands simultaneously. Format disk storagepool volumes sequentially, one at a time, if they are placed into the same filesystem. This should create files which have only a few gaps and will improve sequential read/write performance.

dsmserv.opt
Recommended Options
BUFPoolsize (see Table 1)

LOGPoolsize 512
TXNGroupmax 256
MOVEBatchsize 1000
MOVESizethresh 2048
TCPWindowsize 63
USELARGEBuffer Yes

Solaris Server

Performance Recommendations

The following recommendations may help you to provide optimized TSM performance for your TSM in Solaris environment.

- Use raw partitions for server database, recovery log, and disk storage pool volumes on the Solaris platform. Customer experience and our measurements in the lab show that raw logical volumes offer better client backup/restore throughput and server administrative process performance than UFS or Veritas file system volumes for Solaris.
- The VxFS file system with direct I/O offers I/O characteristics similar to raw I/O on raw partitions. By switching to raw or direct I/O and by giving enough memory, a much larger working set of data can be cached, and a much higher cache hit rate can be sustained with obvious performance benefits. And when physical I/O is required, the CPU cost of performing that I/O is reduced because the data is not first copied to file system buffers. We therefore recommend using VxFS file systems mounted with the direct I/O option (convosync=direct)
- When UFS filesystem volumes are used, mount these filesystems using the **forcedirectio** flag If the filesystem is mounted using forcedirectio, then data is transferred directly between user address space and the disk. If the filesystem is mounted using noforcedirectio, then data is buffered in kernel address space when data is transferred between user address space and the disk. Forcedirectio is a performance option that benefits only from large sequential data transfers. The default behavior is noforcedirectio.
- Formatting disk storagepool volumes which will be placed in the same filesystem with parallel running dsmfmt processes can cause the storagepool volumes to be highly fragmented. For disk storage pool volumes in the same filesystem, do not issue multiple dsmfmt or DEFINE VOLUME commands simultaneously. Format disk storagepool volumes sequentially, one at a time, if they are placed into the same filesystem. This should create files which have only a few gaps and will improve sequential read/write performance.
- Use the TSM server option TXNGroupmax 256 in dsmserv.opt and the TSM client option TXNBytelimit 25600 in dsm.sys to maximize the size of the TSM server file aggregates. File aggregation provides throughput improvements for

many server data movement and inventory functions, such as storage pool migration, storage pool backup, delete filespace/volume and inventory expiration.

- To avoid any Disk I/O contention between TSM server database, recovery log and disk storage pool volumes, it is also essential to use separate physical disk for database, recovery log and disk storage pool.
- Place the Tivoli Storage Manager server database volumes on the fastest disks. If the write cache exists for the disk adapter that the Database volume is attached to, enable it for the best Database performance, especially if the database volumes are mirrored..
- Tune the size of the Tivoli Storage Manager server database buffer pool appropriately according to the total amount of installed system memory, as shown in Table 1.
- For optimal Tivoli Storage Manager backup and restore performance with a large number of clients, consider using Tivoli Storage Manager client compression. Compressing the data on the client reduces demand on the network and on the Tivoli Storage Manager server. The reduced amount of data on the server continues to provide performance benefits whenever this data is moved, such as for storage pool migration and storage pool backup. However, client compression will significantly reduce the performance of each individual client, and the reduction will be more pronounced on the slowest client systems.

dsmserv.opt
Recommended Options
BUFPoolsize (see Table 1)
LOGPoolsize 512
TXNGroupmax 256
MOVEBatchsize 1000
MOVESizethresh 2048
TCPWindowsize 63
USELARGEBuffer Yes
TCPNODELAY Yes

AIX Server

Performance Recommendations

The following recommendations may help you provide optimized Tivoli Storage Manager performance for your environment.

- Use raw partitions for server database, recovery log, and disk storage pool volumes on the AIX platform. Customer experience and measurements in the lab show that raw logical volumes offer better client backup/restore throughput and server administrative process performance.

- If you choose not to use raw logical volumes, direct I/O should always be used for jfs2 file systems. For jfs file systems, direct I/O may cause degradation, especially with large-file enabled file systems.
- Formatting disk storagepool volumes which will be placed in the same filesystem with parallel running dsmfmt processes can cause the storagepool volumes to be highly fragmented. For disk storage pool volumes in the same filesystem, do not issue multiple dsmfmt or DEFINE VOLUME commands simultaneously. Format disk storagepool volumes sequentially, one at a time, if they are placed into the same filesystem. This should create files which have only a few gaps and will improve sequential read/write performance.
- Use the Tivoli Storage Manager server option txngroupmax 256 and the Tivoli Storage Manager client option txnbytelimit 25600 to maximize the transaction size. A larger transaction size increases the size of the Tivoli Storage Manager server file aggregates. File aggregation provides throughput improvements for many server data movement and inventory functions, such as storage pool migration, storage pool backup, and inventory expiration. A larger transaction size when using backup directly to tape reduces the number of tape buffer flushes and therefore can improve throughput.
- Use the Tivoli Storage Manager server option uselargebuffers yes (the default) to improve throughput and server efficiency for large files.
- For optimal Tivoli Storage Manager backup and restore performance with a large number of clients, consider using Tivoli Storage Manager client compression. Compressing the data on the client reduces demand on the network and on the Tivoli Storage Manager server. The reduced amount of data on the server continues to provide performance benefits whenever this data is moved, such as for storage pool migration and storage pool backup. However, client compression will significantly reduce the performance of each individual client, and the reduction will be more pronounced on the slowest client systems.
- If you have new generation SP nodes, we recommend TCPW of 128K for both SP client and SP AIX server. This is especially true if you have ATM card in the SP machine. On the newer and faster SP machines (and faster HPS) TCPW of 128K seems to work well.
- To avoid any Disk I/O contention between TSM server database, recovery log and disk storage pool volumes, it is also essential to use separate physical disk for database, recovery log and disk storage pool.
- Place the Tivoli Storage Manager server database volumes on the fastest disks. If the write cache exists for the disk adapter and that adapter has attached disks that contain only Database volumes (no Storage Pool volumes) then enable the write cached for the best Database performance.
- Tune the size of the Tivoli Storage Manager server database buffer pool appropriately according to the total amount of installed system memory, as shown in Table 1.

dsmserv.opt
Recommended Options

BUFPoolsize (see Table 1)
LOGPoolsize 512
TXNGroupmax 256
MOVEBatchsize 1000
MOVESizethresh 2048
TCPWindowsize 63 (see the recommendation above for SP nodes)
USELARGEBuffer Yes
TCPNODELAY Yes

Linux Server

Performance Recommendations

The following recommendations may help you provide optimized Tivoli Storage Manager performance for your environment.

- Formatting disk storagepool volumes which will be placed in the same filesystem with parallel running dsmfmt processes can cause the storagepool volumes to be highly fragmented. For disk storage pool volumes in the same filesystem, do not issue multiple dsmfmt or DEFINE VOLUME commands simultaneously. Format disk storagepool volumes sequentially, one at a time, if they are placed into the same filesystem. This should create files which have only a few gaps and will improve sequential read/write performance.
- Use the Tivoli Storage Manager server option txngroupmax 256 and the Tivoli Storage Manager client option txnbytelimit 25600 to maximize the transaction size. A larger transaction size increases the size of the Tivoli Storage Manager server file aggregates. File aggregation provides throughput improvements for many server data movement and inventory functions, such as storage pool migration, storage pool backup, and inventory expiration. A larger transaction size when using backup directly to tape reduces the number of tape buffer flushes and therefore can improve throughput.
- Use the Tivoli Storage Manager server option uselargebuffers yes (the default) to improve throughput and server efficiency for large files.
- Disable any unneeded Daemons (Services). Most enterprise distributions come with an incredible amount of features, however most of the time only a small subset of these features will be used. For example, the TCPIP data movement could be blocked or slowed down significantly by the internal firewall in SUSE 9 x86_64 (It could be terminated by /etc/init.d/SuSEfirewall2_setup stop).
- For optimal Tivoli Storage Manager backup and restore performance with a large number of clients, consider using Tivoli Storage Manager client compression. Compressing the data on the client reduces demand on the network and on the Tivoli Storage Manager server. The reduced amount of data on the server continues to provide performance benefits whenever this data is moved, such as for storage pool migration and storage pool backup. However, client compression

will significantly reduce the performance of each individual client, and the reduction will be more pronounced on the slowest client systems.

- To avoid any Disk I/O contention between TSM server database, recovery log and disk storage pool volumes, it is also essential to use separate physical disk for database, recovery log and disk storage pool.
- Place the Tivoli Storage Manager server database volumes on the fastest disks. If the write cache exists for the disk adapter and that adapter has attached disks that contain only Database volumes (no Storage Pool volumes) then enable the write cached for the best Database performance.
- Tune the size of the Tivoli Storage Manager server database buffer pool appropriately according to the total amount of installed system memory, as shown in Table 1.

dsmserv.opt
Recommended Options
BUFPoolsize (see Table 1)
LOGPoolsize 512
TXNGroupmax 256
MOVEBatchsize 1000
MOVESizethresh 2048
TCPWindowsize 63
USELARGEBuffer Yes
TCPNODELAY Yes

HP-UX Server

Performance Recommendations

The following recommendations may help you provide optimized Tivoli Storage Manager performance for your environment.

- Use raw partitions for Disk Storage Pool on HP-UX TSM server. Raw partitions are recommended because our measurements in the lab show that raw partition volumes offer better backup/restore throughput than when VXFS volumes are used on HP-UX..
- Formatting disk storagepool volumes which will be placed in the same filesystem with parallel running dsmfmt processes can cause the storagepool volumes to be highly fragmented. For disk storage pool volumes in the same filesystem, do not issue multiple dsmfmt or DEFINE VOLUME commands simultaneously. Format disk storagepool volumes sequentially, one at a time, if they are placed into the same filesystem. This should create files which have only a few gaps and will improve sequential read/write performance.

- Use the Tivoli Storage Manager server option txngroupmax 256 and the Tivoli Storage Manager client option txnbytelimit 25600 to maximize the transaction size. A larger transaction size increases the size of the Tivoli Storage Manager server file aggregates. File aggregation provides throughput improvements for many server data movement and inventory functions, such as storage pool migration, storage pool backup, and inventory expiration. A larger transaction size when using backup directly to tape reduces the number of tape buffer flushes and therefore can improve throughput.
- Use the Tivoli Storage Manager server option uselargebuffers yes to improve throughput and server efficiency for large files.
- Place the Tivoli Storage Manager server database volumes on the fastest disks. If the write cache exists for the disk adapter that the Database volume is attached to, enable it for the best Database performance.
- Tune the size of the Tivoli Storage Manager server database buffer pool appropriately. The Tivoli Storage Manager database buffer pool acts as a cache for database pages, so when a page is needed, and the page already exists in the buffer pool, an I/O can be avoided. After establishing a significant Tivoli Storage Manager server workload, tuning the database buffer pool size should be done using the cache hit percentage value provided by the Tivoli Storage Manager server QUERY DB command. Also observe the operating system paging statistics (via iostat, vmstat, sam) to be sure that additional memory requirements are not causing excessive paging.
- For optimal Tivoli Storage Manager backup and restore performance with a large number of clients, consider using Tivoli Storage Manager client compression. Compressing the data on the client reduces demand on the network and on the Tivoli Storage Manager server. The reduced amount of data on the server continues to provide performance benefits whenever this data is moved, such as for storage pool migration and storage pool backup. However, client compression will significantly reduce the performance of each individual client, and the reduction will be more pronounced on the slowest client systems.
- To avoid any Disk I/O contention between TSM server database, recovery log and disk storage pool volumes, it is also essential to use separate physical disks for database, recovery log and disk storage pool.

dsmserv.opt
Recommended Options
BUFPoolsize (see Table 1)
LOGPoolsize 512
TXNGroupmax 256
MOVEBatchsize 1000
MOVESizethresh 2048
TCPWindowsize 63

USELARGEBuffer Yes
TCPNODELAY Yes

TSM Client Tuning

This chapter focuses on tuning the TSM client to obtain maximum performance when using TSM. The first section is a discussion of the tuning parameters applicable to all TSM clients. The second section discusses general client performance considerations. It is then followed by platform specific recommendations..

Tuning Options

The following parameters are tunable on all TSM clients for all available releases:

- COMPRESSION
- COMPRESSALWAYS
- COMMRESTARTDURATION / COMMRESTARTINTERVAL
- QUIET
- DISKBUFFSIZE
- LARGECOMmbuffers
- MEMORYEFFICIENTBACKUP (previously SLOWINCREMENTAL)
- Multi-Session Restore
- RESOURCEUTILIZATION
- TAPEPrompt
- TCPBuffsize
- TCPNodelay
- TCPWindowsize
- TXNBytelimit

Note: On UNIX clients the TSM client options are put in either the DSM.SYS file or the DSM.OPT file.

COMPRESSION

Specifies if compression is enabled on the TSM client. The default is No.

For optimal Tivoli Storage Manager backup and restore performance with a large number of clients, consider using Tivoli Storage Manager client compression. Compressing the data on the client reduces demand on the network and on the Tivoli Storage Manager server. The reduced amount of data on the server continues to provide performance benefits whenever this data is moved, such as for storage pool migration and storage pool backup. However, client compression may significantly reduce the performance of each individual client--the reduction will be more pronounced on the slowest client systems.

For maximum performance with a single fast client, fast network and fast server, turn compression off.

Two alternatives exist to using TSM compression:

- If you are backing up to tape, and the tape drive supports its own compaction, use the tape drive compaction (Refer to "Tape Drives: Compaction" for the appropriate guidelines).
- Do not use TSM compression if a client currently has built-in file compression support. TSM compression on these clients will not yield additional reduction in the amount of data backed up to the server. The following platforms have optional built-in file compression: NetWare, Windows.

Compression can cause severe performance degradation when there are many retries due to failed compression attempts. Compression fails when the compressed file is larger than the original. The TSM client detects this and will stop the compression, fail the transaction and resend the entire transaction uncompressed. This occurs because the file type is not suitable for compression or the file is already compressed (zip files, tar files, etc.). Short of turning off compression, there are two options you can use to reduce or eliminate retries due to compression:

- Use the COMPRESSALWAYS option (see below). This option will eliminate retries due to compression.
- Use the EXCLUDE.COMPRESSION option in the client options file. This option will disable compression for specific files or sets of files. Look in the client output (dsmsched.log) for files that are causing compression retries and then filter those file types.

Recommendation
COMPression No (single fast client, fast network, fast server)
COMPression Yes (multiple clients, slow network, slow server)

COMPRESSALWAYS

The compressalways option specifies whether to continue compressing an object if it grows during compression, or resend the object, uncompressed. This option is used with the compression option.

The compressalways option is used with the archive, incremental, and selective commands. Note: This option can also be defined on the server. If compressalways yes (the default) is specified, files continue compression even if the file size increases. To stop compression if the file size grows, and resend the file, uncompressed, specify compressalways No. This option controls compression only if your administrator specifies that your client node determines the selection.

To reduce the impact of retries, use compressalways yes.

Recommendation
COMPRESSAlways Yes

COMMRESTARTDURATION / COMMRESTARTINTERVAL

To make clients are more tolerant of network connectivity interruptions, use the options COMMRESTARTDURATION and COMMRESTARTINTERVAL to control the restart window of time and interval between restarts. These options assist in environments which are subject to heavy network congestion or frequent interruptions, and eases the manageability of large numbers of clients by reducing intervention on error conditions. In a sense, performance is improved if consideration is given to account for time to detect, correct, and restart as a result of an error condition.

Note: A scheduled event will continue if the client reconnects with the server before the commrestartduration value elapses, even if the event's startup window has elapsed."

Syntax
COMMRESTARTDURATION minutes COMMRESTARTINTERVAL seconds
minutes The maximum number of minutes you want the client to attempt to reconnect with a server after a communication failure occurs. The range of values is zero through 9999; the default is 60. **seconds** The number of seconds you want the client to wait between attempts to reconnect with a server after a communication failure occurs. The range of values is zero through 65535; the default is 15.

Recommendation
COMMRESTARTDURATION (Tune to your environment)
COMMRESTARTINTERVAL (Tune to your environment)

QUIET

Specifies the QUIET option. The default is VERBOSE.

The QUIET option keeps messages from being written to the screen during TSM backups. By default, TSM displays information about each file it backs up. To prevent this, use the QUIET option. However, messages and summary information are still written to the log files.

Two main benefits exist when using the QUIET option:

- For tape backup, the first transaction group of data is always resent. To avoid this, use the QUIET option to reduce retransmissions at the client.
- If you are using the client scheduler to schedule TSM backups, using the QUIET option dramatically reduces disk I/O overhead to the schedule log and improves TSM throughput.

Recommendation
Quiet

DISKBUFFSIZE

Specifies the maximum disk I/O buffer size (in kilobytes) that the client may use when reading files.

NOTE: The DISKBUFFSIZE option replaces the LARGECOMMBUFFERS option.

Optimal backup, archive, or HSM migration client performance may be achieved if the value for this option is equal to or smaller than the amount of file read ahead provided by the client file system. A larger buffer will require more memory and may not improve performance.

The default value is 32 for all clients except AIX. For AIX, the default value is 256 except when ENABLELANFREE YES is specified. When ENABLELANFREE YES is specified on AIX, the default value is 32.

Recommendation
DISKBUFFSIZE (use the default value)

LARGECOMmbuffers

- NOTE: For TSM 5.3 and later clients, this option has been replaced by the ***diskbuffsize*** option. At this time, ***largecommbuffers*** will continue to be accepted by the TSM client in order to ease the transition to the new option. However, the value specified by ***largecommbuffers*** will be ignored in favor of the ***diskbuffsize*** setting.
- The LARGECOMmbuffers option specifies whether the client uses increased buffers to transfer large amounts of data between the client and the server. This option is used to increase the disk I/O buffers from 32KB to 256 KB for a client reading or writing data from its disk. This client option, formally called USELARGEbuffer, has been renamed to LARGECOMmbuffers to avoid confusion with the server option.
- Need to tune filesystem read ahead to get benefits

- Can't tune read ahead on most platforms
- Optimal AIX TSM client performance in the lab (with LARGECOMmbuffers YES) was achieved by increasing the amount of AIX disk read-ahead. (See "AIX's VMTUNE")
- You can disable this option when your workstation is running low on memory.
- Never use LARGECOMmbuffers YES with LANFree operations

Recommendation
LARGECOMmbuffers Yes (for AIX only)
LARGECOMmbuffers No (All platforms except AIX and all platforms with LANfree)

MEMORYEFFICIENTBACKUP (previously SLOWINCREMENTAL)

Specifies if the MEMORYEFFICIENTBACKUP option is enabled on the TSM client. The default is No.

With the MEMORYEFFICIENTBACKUP option, TSM backs up one directory at a time. By default, TSM backs up all the directories at once. The default option is very memory intensive. To save memory on clients with a limited amount of memory and large number of directories and/or files, turn MEMORYEFFICIENTBACKUP on.

The amount of memory required by the TSM client can be substantial when performing incremental backups of large filesystems. We generally use the rule of about 300 bytes of inventory data per file, so that a filesystem of 1 million files would require about 300MB of virtual memory in which to hold the server inventory data used to determine which files must be backed up. This memory requirement can be reduced by using the TSM client option MemoryEfficientBackup Yes, However, this can be expected to impact the incremental backup elapsed time significantly. Some alternatives:

- Partition the filesystem and process each partition sequentially – Use the virtualmountpoint option (unix only)
- Use Include/Exclude list to backup only what is necessary
- Use multiple filesystems
- Use Journal Based Backup (Windows only)

Recommendation
MEMORYEFFICIENTBACKUP No

Multi-Session Backup/Restore

IBM Tivoli Storage Manager 5.1 introduced Multi-Session restore which allows the backup-archive clients to perform multiple restore sessions for no-query restore operations, increasing the speed of restores. This is similar to the multiple backup session support. Please refer to the Tivoli Storage Manager Backup-Archive Installation

and User's Guide, for each supported client platform, for more detailed information about the no-query concept.

The Multi-Session restore exploits the mount points available on the server. If the data to be restored resides on several tapes, and there are sufficient mount points available, and the restore is done using the no-query restore protocol, then multiple sessions can be used to restore the data. If there are not enough mount points available, only the client, not the server, will issue the ANS1312E message.

The number of sessions used during the restore is dependant on the number of mount points available, the MAXNUMMP setting for the client node in the server, and the RESOURceutilization parameter setting in the client options file. Since the number of sessions increase when performing the Multi-Session restore, please use a proper setting for the dsmserv.opt file parameter MAXSessions. See Resourceutilization for more information.

Some Multi-Session restore considerations:

- Only one session used when restoring from random access disk storage pools
- Only one filesystem can be restored at a time with the command line – multiple sessions may still be used for a single filesystem
- Even small clients can gain throughput for restores requiring many tape mounts or locates
- Watch out for tape cartridge contention especially when restoring from a collocated node

Some Multi-Session backup considerations:

- Only one session will compare attributes for incremental backup. Incremental backup throughput does not improve for a single filesystem with a small amount of changed data
- Data transfer sessions do not have filesystem affinity, so each session could send files from multiple filesystems. This is good for workload balancing. This is not so good if backing up direct to tape storage pool collocated by filespace – do not use multisession client to backup directly to a storage pool collocated by filespace – use multiple commands, one per filespace
- Multiple sessions may not get started if there are not enough entries on the transaction queue

RESOURCEUTILIZATION

Resourceutilization is a client option used to regulate the level of resources (i.e. number of concurrent sessions) the TSM client and server can use during processing. Multiple sessions are created automatically through a single invocation of a TSM backup/restore or archive/retrieve command.

Even though the multi-session function is transparent to the end user and is started automatically, there are some parameters which enable the user to customize the function.

The way to influence the multi-session function is with the option RESOURCEUTILIZATION. This option increases or decreases the ability of the Tivoli Storage Manager client to create multiple sessions. For Backup or Archive the value of RESOURCEUTILIZATION does not directly specify the number of sessions created by the client. However, this setting specifies the level of resources the Tivoli StorageManager server and client can use during backup or archive processing. The higher the value, the more sessions the client can start if it deems necessary. The range for the parameter is from 1 to 10. When the option is not set, which is the default, then only two sessions are created to the server.

The default resourceutilization level allows up to two sessions with the server, one for querying the server and one for sending file data. Resourceutilization=5 permits up to four sessions (two for queries and two for sending data), while resourceutilization=10 permits up to eight sessions (four for queries and four for sending data) with the server. The relationship between Resourseutilization and the maximum number of sessions created is part of an internalized algorithm and, as such, is subject to change. This table lists the relationships between Resourceutilization values and the maximum sessions created. Producer sessions scan the client system for eligible files. The remaining sessions are consumer sessions and are used for data transfer. The threshold value affects how quickly new sessions are created.

Table 2. Resource Utilization Settings

RU Value	Max # of Sessions	Unique # of Producer Sessions	Threshold (sec)
1	1	0	45
2	2	1	45
3	3	1	45
4	3	1	30
5	4	2	30
6	4	2	20
7	5	2	20
8	6	2	20
9	7	3	20
10	8	4	10
default (0)	2	1	30

Backup throughput improvements that can be achieved by increasing the Resourceutilization level will vary from client node to client node. Factors that affect throughputs of multiple sessions include the configuration of the client storage subsystem (i.e. layout of file systems on physical disks), the client's ability to drive multiple sessions (sufficient CPU, memory), the server's ability to handle multiple client sessions (CPU, memory, number of storage pool volumes), and sufficient bandwidth in the network to handle the increased traffic.

The MAXSESSIONS parameter controls the maximum number of simultaneous client sessions with the Tivoli Storage Manager server. The total number of parallel sessions for a client is counted for the maximum number of sessions allowed with the server. For this reason, you need to consider whether to increase the value of the MAXSESSIONS parameter in the server option file.

When using the Resourceutilization option to enable multiple client/server sessions for backup direct to tape, the client node maximum mount points allowed parameter, MAXNUMMP, must also be updated at the server (using the Update Node command).

If the TSM client has a filesystem spread across multiple disks (RAID 0 or RAID 5), or multiple large filesystems, we recommend setting the TSM client option Resourceutilization to a value of 5 or 10. This will enable multiple sessions with the server during backup or archive, and can result in substantial throughput improvements in some cases. It is not likely to improve incremental backup of a single large filesystem with a small percentage of changed data. If backup is direct to tape, then the client node maximum mount points allowed parameter, MAXNUMMP, must also be updated at the server using the update node command.

When a restore is requested, the default is to use a maximum of two sessions, based on how many tapes the requested data is stored on, how many tape drives are available, and the maximum number of mount points allowed for the node.

The default value for the RESOURceutilization parameter is one (1) and the maximum value is ten (10).

For example, if the data to be restored are on five different tape volumes, and the maximum number of mount points is five for the node requesting the restore, and RESOURceutilization is set to three, then three sessions will be used for the restore. If the RESOURceutilization setting is increased to five, then five sessions will be used for the restore. There is a one-to-one (1:1) relationship to the number of restore sessions allowed and the RESOURceutilization setting.

Resourceutilization Values

- Values >2 allowed only when connecting to a >3.7 server
- Non-root UNIX limited to 1 session

Recommendation
RESOURceutilization 1 for workstations, 5 for a small server, 10 for a large server

TAPEPrompt

Specifies if the user wants to be prompted for tape mounts.

The TAPEPrompt option specifies if you want TSM to wait for a tape to be mounted for a backup, archive, restore or retrieve operation, or prompt you for your choice.

Recommendation
TAPEPrompt No

TCPBufsize

The TCPBuffsize option specifies the size of the internal TCP communication buffer, that is used to transfer data between the client node and the server. A large buffer can improve communication performance, but requires more memory.

Specifies the size of the buffer in kilobytes. The default is 32KB, and the maximum is now 512KB.

Recommendation
TCPBufsize 32

TCPNodelay

Specifies the operating system's TCPNodelay parameter is set to Yes or No. The default is Yes.

The TCPNodelay option, when set to Yes, disables the TCP/IP Nagle algorithm which allows data packets less than the MTU size to be sent out immediately. This generally results in better performance for TSM client/server communications.

Recommendation
TCPNodelay Yes

Note: This option is also valid on the TSM server.

TCPWindowsize

This option specifies the size of the TCP sliding window in kilobytes. The TCPWindowsize option overrides the operating system's TCP send and receive spaces. In AIX, for instance, these parameters are set as “no” options tcp_send space and tcp_recvspace. Specifying TCPWindowsize 0, will cause TSM to use the operating system default. This is not recommended because TSM’s optimal setting might not be same as the optimal setting for other applications. The default is 63KB, and the maximum is 2048KB. TCPWindowsize option specifies the size of the TCP sliding window for all clients and all but MVS servers. A larger window size can improve communication performance, but uses more memory. It enables multiple frames to be sent before an acknowledgment is obtained from the receiver. If long transmission delays are being observed, increasing the TCPWindowsize may improve throughput.

Recommendation #1
TCPWindowsize 63

Recommendation #2 (SP2 environment with a high speed switch)
TCPWindowsize 128 - 640

Note: This option is also valid on the TSM server.

TXNBytelimit

The txnbytelimit option specifies the maximum transaction size in kilobytes for data transferred between the client and server. The range of values is 300 through 2097152 (2 GB); the default is 25600. A transaction is the unit of work exchanged between the client and server. Because the client program can transfer more than one file or directory between the client and server before it commits the data to server storage, a transaction can contain more than one file or directory. This is called a transaction group. This option permits you to control the amount of data sent between the client and server before the server commits the data and changes to the server database, thus affecting the speed with which the client performs work. The amount of data sent applies when files are batched together during backup or when receiving files from the server during a restore procedure. The server administrator can limit the number of files or directories contained within a group transaction using the txngroupmax option (Refer to "TXNGroupmax"). The actual size of a transaction can be less than your limit. Once this number is reached, the client sends the files to the server even if the transaction byte limit is not reached.

There are several items to consider when setting this parameter:

- Increasing the amount of data per transaction will increase recovery log requirements on the server. Check log and log pool space to ensure there is enough space. Also note that a larger log may result in longer server start-up times.
- Increasing the amount of data per transaction may result in more data being retransmitted if a retry occurs. This may negatively affect performance.

- The benefits of changing this parameter are dependent on configuration and workload characteristics. In particular, this parameter benefits tape storage pool backup more so than disk storage pool backup, especially if many small files are in the workload.

When setting the size of transactions consider setting a smaller size if you are suffering many resends due to files changing during backup when using static, shared static, or shared dynamic. This would apply to static as well as to shared because when the client realizes a file has changed during a backup and decides to not send it, the file that is, it would still have to re-send the other files in that transaction. To enhance performance with TSM and LTO or DLT tape drives, be sure to set TXNBytelimit to the max 2097152, and on the server, raise TXNGroupmax to 256. Additionally, for small file workloads, first stage the backups to a disk storage pool and then migrate to LTO.

Recommendation (V4)
TXNBytelimit 25600 TXNBytelimit 2097152 (with LTO or DLT)

General Performance Considerations

Backup/Restore Performance: Multi-Client Backups/Restores

Two or more clients on the same workstation may give better throughputs for backup/restore. This is beneficial for workstations with large file systems on multiple disks. You may consider running a backup/restore for each filespace. However, the tradeoffs for running multiple backups/restores are increased memory usage and CPU utilization.

Network Traffic: Reducing Data Flow from the Client

Use INCLUDE/EXCLUDE lists to eliminate unnecessary backups. The use of virtual mountpoints will also limit the amount of data that must be backed up.

Overhead: Client Processing

Using the -FROMNODE option creates additional overhead on all clients. Consider using the VIRTUALNODENAME instead of the –FROMNODE option.

Platform Specific Recommendations

AIX Client

dsmsys.opt
Recommended Options

tcpwindowsize 63 (SP2 128-640)
tcpbufsize 32
largecommbuffers yes (no if using LAN-free)
Diskbuffsize (don't specify, use the default)
txnbytelimit 25600
RESOURCEUtilization (tune to your environment)

Linux Client

dsmsys.opt
Recommended Options
tcpwindowsize 63
tcpbufsize 32
largecommbuffers no
Diskbuffsize (don't specify, use the default)
txnbytelimit 25600
RESOURCEUtilization (tune to your environment)

Windows Client

Tuning Tips

- Keep in mind that using NTFS on the TSM client does cause a performance degradation for small file workloads relative to FAT, because of the need to check security information during backup and the impact of the NTFS filesystem recovery capability during restore.
- NTFS file compression should not be used on disk volumes that are used by the TSM server, because of the potential for performance degradation.
- For optimal TSM backup and restore performance when using a local client, use the namedpipes communication method.
- For optimal TSM backup and restore performance with a large number of clients, consider using TSM client compression. Compressing the data on the client reduces demand on the network and on the TSM server. The reduced amount of data on the server continues to provide performance benefits whenever this data is moved, such as for storage pool migration and storage pool backup. However, client compression will significantly reduce the performance of each individual client, and the reduction will be more pronounced on the slowest client systems.
- Anti-virus software can negatively impact backup and restore performance

dsm.opt

Recommended Options
tcpwindowsize 63
tcpbufsize 32
largecommbuffers no
Diskbuffsize (don’t specify, use the default)
txnbytelimit 25600
RESOURceutilization (tune to your environment)

Windows Journal Based Backup

Instead of cross referencing current state of files with the TSM database we just backup files indicated as changed in the change journal. Uses a real-time determination of changed files/directories and avoids filesystem scan and attribute comparison. It is much faster than classic incremental but improvement depends on amount of changed data. Also requires less memory usage and less disk usage.

Requires the installation of a Tivoli Journal Engine Service which monitors filesystems activity for file changes. Impacts performance of the filesystem slightly. (On the order of 5% during a Netbench test run.)

Especially good for large filesystems with many files that don't change often.

Journal options specified in tsmjbbd.ini. Defaults work well, just add the filesystems to be monitored.

Apple Macintosh OS Client

Tuning Tips

- For best performance, try to limit the use of Extended Attributes or, if Extended Attributes are used, limit their length.
- For optimal TSM backup and restore performance with a large number of clients, consider using TSM client compression. Compressing the data on the client reduces demand on the network and on the TSM server. The reduced amount of data on the server continues to provide performance benefits whenever this data is moved, such as for storage pool migration and storage pool backup. However, client compression will significantly reduce the performance of each individual client, and the reduction will be more pronounced on the slowest client systems.
- Anti-virus software can negatively impact backup and restore performance

System Preferences
Recommended Options
tcpwindowsize 63

tcpbufsize 32
largecommbuffers no
txnbytelimit 25600
RESOURceutilization (tune to your environment)

DEC Client

On DEC ULTRIX, the parameter that most affects performance is Maximum Transmission Unit (MTU). On Ethernet LANs, the largest MTU value is 1500.

- Maximum Transmission Unit

The ifconfig command can be used to change the MTU size value. If the adapter receive buffer sizes can be configured to larger values, increasing the MTU size on the workstation is recommended. As discussed earlier if the frame is going through networks with smaller MTU's, increasing the MTU size may degrade performance because of overhead encountered in disassembling packets by routers and assembling packets at the destination.

Novell Client

PROCESSORutilization

Specifies in hundredths of a second the length of time TSM controls the CPU. The default is 1. The minimum is 0, and there is no maximum value.

If set to less than 1, this parameter could have a negative impact on performance. Increasing this value will increase TSM's priority to the CPU, lessening other process's priorities. Setting PROCESSORutilization greater than 20 may prevent other scheduled processes or NetWare requestors from accessing the file server.

Recommendation
PROCESSORutilization 1 - 20

dsm.opt
Recommended Options
tcpwindowsize 643
tcpbufsize 32
largecommbuffers no
Diskbuffsize (don’t specify, use the default)
txnbytelimit 25600

SOLARIS Client

On SunOS, the TCP/IP software parameters can be changed by editing the min inetinet/in_proto.c file in the release kernel build directory (usually/usr/sys). After changing the parameters, you need to rebuild the kernel. The parameters that can affect performance are:

tcp_default_mss

Specifies the default Maximum Segment Size (MSS) for TCP in bytes. The MSS is based on the Maximum Transmission Unit (MTU) size of the network if the destination is on the same network. To avoid fragmentation, the conservative value of 512 is used . For improved performance on Ethernet or Token-Ring, larger MSS values are recommended. (For example, settings of 1024, 1500, 2048, or 4096 can be used.) On Ethernet LANs, the largest MTU value is 1500.

tcp_sendspace

Specifies the number of bytes that the user can send to a TCP socket buffer before being blocked. The default values can be changed on a given socket with the SO_SNDBUF ioctl. The default value is 4096

Recommendation
Set this parameter to 16KB or 32KB.

tcp_recvspace

Specifies the number of bytes that the remote TCP can send to a TCP socket buffer before being blocked. The default values can be changed on a given socket with the SO_RCVBUF ioctl. The default value is 4096.

Recommendation
Set this parameter to 16KB or 32KB.

TSM Options in the DSM.SYS File

The following parameters impact performance for the backup and restore functions in TSM:

dsmsys.opt
Recommended Options
tcpwindowsize 63
tcpbufsize 32
largecommbuffers no
Diskbuffsize (don’t specify, use the default)
txnbytelimit 25600

TCP buffers tcp_xmit_hiwat and tcp_recv_hiwat must match your tcpwindowsize.

Do not automatically use the TSM client option LargeCommbuffers yes. The default setting for this option is NO for TSM client for Solaris. Some experiments may be required to determine which value of this option in the client is appropriate. Some specific client workload tests, for example, using LargeCommbuffers NO provided better throughput. Issues concerning the amount of disk file fragmentation, the amount of file read ahead performed by the operating system, the average client file size, the size of the physical memory in the client and the type and speed of the disk drives in use need to be taken into account in determining the appropriate setting for this option..

Command Line Options

A group of options exists which can be used with specific commands on the command line only. When specifying options with a command, always precede the option with a dash (-). Two command line options which may improve TSM performance are:

- IFNEWER
- INCRBYDATE

It is also important to note here that the command line interface is generally faster than the GUI and requires less overhead.

IFNEWER

This option is used in conjunction with the restore command, and restores files only if the server date is newer than the date of the local file. This option may result in lower network utilization if less data must travel across the network.

INCRBYDATE

In a regular incremental backup, the server reads the attributes of all the files in the filesystem and passes this information to the client. The client then compares the server list to a list of its current file system. This comparison can be very time-consuming, especially on clients on the following platforms: NetWare, Macintosh and Windows. These clients sometimes have a limited amount of memory.

With an incremental by date backup, the server only passes the date of the last successful backup. Now it is no longer necessary to query every active file on the TSM server. The time savings are significant. However, periodic regular incrementals are still needed to backup files that have only had their attributes changes. For example: If a new file in your file system has a creation date previous to the last successful backup date. All future INCRBYDATEs will not backup this file, because the client will think it has already been backed up. Also, files that have been deleted are not detected by the INCRBYDATE backup. These deleted files will be restored if you perform a full system restore.

Chapter 3. TSM Optimization

Server Optimization

- Database Performance
- Disk Use
- Backup Performance:Number of Backup Versions
- Disaster Recovery:IMPORT/EXPORT vs. Storage Pool Backup/Restore
- Cached Disk Storage Pool
- Migration: Tuning Number of Migration Processes
- Migration: Tuning Migration Thresholds
- Tape Drives: Collocation
- TSM HSM Tuning Considerations
- Problem Determination: Server Activity Log and Other System Logs
- Server Activity: Scheduling of TSM Sessions and Processes

Database Performance

Database mirroring provides higher reliability, but comes at a cost in performance (this is especially noticed with sequential mirroring). To minimize the impact of database write activity, use disk subsystems with non-volatile write caching ability (I.e. ESS 2105, SSA Fast/write cache (for AIX), ServerRAID (for Intel,) etc.). Use of disks (and/or disk adaptors) with write cache can nearly double (or more) the performance of database write activity (when parallel mirroring.). This is true even if there appears to be plenty of bandwidth left on the database disks.

Disk Use

Server Configuration – Disk: Use as many independent physical disks as you can afford to minimize I/O contention. Configure one TSM volume per physical disk, or at most two. Separate the recovery log, database, disk storage pool volumes. Place TSM volumes at the outside diameter of physical disk. This gives better sequential throughput and faster seek time. Using raw logical volumes on Unix systems can cut CPU consumption but may be slower during storage pool migrations. (Due to lack of read ahead)

Choosing physical disks (JBOD) or RAID arrays: RAID will require more physical disks for equivalent performance . Be sure to consider the write penalty of RAID5 arrays. Write throughput is important during backup/archive. TSM recovery log and database mirroring provides better recoverability than hardware redundancy.

Server Configuration - Disk Write Cache: Use disk subsystem/adapter write cache for all RAID 5 arrays and physical disks with TSM database volumes (random I/O). Do not use disk subsystem/adapter write cache for physical disks with TSM storage pool volumes (sequential I/O).

TSM server database: Use multiple physical disks. Divide database size equally over all volumes. Place volumes at disk outer diameter to minimize seek time. Use disk subsystems with non-volatile write cache (ESS 2105, SSA 7133, ServeRAID). Use optimal database bufferpool size. Use MIRRORWRITE DB PARALLEL and use database page shadow to reduce overhead of mirrored writes. Use multiple recovery log and database volumes.

Backup Performance: Number of Backup Versions

If possible, limit the number of versions of any backup file to what is really needed. File backup performance will degrade when there are many versions of an object. Use the DEFINE COPYGROUP command and modify the VEREXISTS parameter to control the number of version, or use the UPDATE COPYGROUP command. The default number of backup versions is 2.

Disaster Recovery: IMPORT/EXPORT vs. Storage Pool Backup/Restore

TSM servers have a facility to backup and restore the storage pools for disaster recovery. Some customers use Import/Export for disaster recovery. The performance of Storage Pool Backup/Recovery is superior to Import/Export for disaster recovery.

Cached Disk Storage Pool

Using cached disk storage pools can increase restore performance by avoiding tape mounts. The benefit is seen for restoring files that were recently backed up. If the disk pool is large enough to hold a day's worth of data, then caching is a good option. Use the DEFINE STGPOOL command and modify the CACHE option to enable caching, or use the UPDATE STGPOOL command. However, when the TSM storage pool is large and Cache Disk Storage Pool has been set to YES, the storage pool may become fragmented and response will suffer. If this condition is suspected, our recommendation is to turn disk storage pool caching off. Also, disk caching can affect backup throughput because TSM database updates are required to delete cached files in order to create space for the backup files. To improve backup performance, turn disk storage pool caching off. To clear your cached files:

- Issue query stgpool <disk stgpoolname> f=d (admin client)
- Be sure the Percent Migratable is zero
- Turn off Caching using UPDate STGpool
- Issue MOVe Data <stgpool_vol_name> without the STGpool=parm. Files will be moved to other volumes within the same storage pool and cached files will be deleted.

Migration: Tuning Number of Migration Processes

Use the DEFINE STGPOOL command and modify the MIGPROCESS parameter to control the number of migration processes that are used for migrating files from a disk

storage pool to a tape storage pool. When data is migrated from disk to tape, multiple processes can be used if multiple tape drives are available. In some cases, this can improve the time to empty the disk storage volumes, since each migration process works on data for different client nodes. For the MIGPROCESS option, you can specify an integer from 1-999, inclusive, but it should not exceed the number of tape drives available. The default value is 1. You can also use the UPDATE STGPOOL command to modify the number of migration processes.

Migration: Tuning Migration Thresholds

Use the DEFINE STGPOOL command and modify the HIGHMIG and LOWMIG parameters to tune migration thresholds. Tuning the migration thresholds may help improve performance. If the thresholds are set too high, migration will be delayed. This can cause the disk storage pool to fill, and when a client attempts to send data to the disk storage pool it will see the full condition and attempt to go to the volume indicated at the next level in the storage hierarchy. If this is a tape pool, then all drives may be in use by a migration process, in which case the client session will wait on the tape media to be freed by the migration process. The client then sits idle. In this case, the migration thresholds should be lowered so migration starts earlier, or more disk space should be allocated to the TSM disk storage pool. You can also use the UPDATE STGPOOL command to modify the migration thresholds.

Network Traffic: Reducing Network Traffic to the Server

There are several TSM server SET commands that can help limit the amount of network traffic due to client sessions. They are as follows:

- SET QUERYSCHEDPERIOD sets the frequency that a client can contact the server to obtain scheduled work (polling mode). This overrides the client setting. A shorter frequency means more network traffic due to polling. Use longer settings (6-12 hours) to reduce network traffic. Alternately, use Server Prompted schedule mode to eliminate network traffic due to polling.
- SET MAXCMDRETRIES sets global limit on number of times a client scheduled command retries. This overrides the client setting. A smaller number of reduce network traffic due to retries.
- SET RETRYPERIOD specifies the number of minutes between a retry of a scheduler after a failed attempt to contact the server. This overrides the client setting. A larger value will reduce network traffic due to retries and will make successful retry more likely. Be sure to consider your schedule start-up windows when setting the MAXCMDRETRIES and RETRYPERIOD. If a retry is attempted outside of the start-up window, it will fail.

Tape Drives: Collocation

Using collocation will significantly improve the performance of restores for large amounts of data with many files, since fewer tapes will be searched for the necessary files.

It also decreases the chance for media contention with other clients. The trade-off is that more tapes will be needed. Use the DEFINE STGPOOL command and modify the COLLOCATE option to enable collocation, or use the UPDATE STGPOOL command. The default for a storage pool is no collocation.

TSM HSM Tuning Considerations

It is possible to manually migrate and recall selected files using HSM. However, you should use this method only when you are working with small numbers of files. The performance of HSM migration will be poor at best for very small files which are grouped together with wildcard invocation of DSMMIGRATE command as an example. This is because HSM works on one file at a time, unlike archive, retrieve, restore and backup which group files at a transaction boundary. There will be one transaction for each file when HSM migration/recall is used. For a group of small files, it is better to use archive or backup to store them to the server.

If you have to migrate a group of small files to the server, you will get better performance if you go to disk rather than tape. Once the files are HSM migrated to disk, then you can use storage pool migration to move the files to tape.

Problem Determination: Server Activity Log and Other System Logs

Many times when performance problems occur, an abnormal system condition may have been the cause. Often the cause of these problems can be determined by examining TSM server activity logs, client error file or the appropriate system logs for your operating system.

Server Activity: Scheduling of TSM Sessions and Processes

TSM throughput can degrade if all client backups are started simultaneously. It is best to avoid concurrent backups, and it would be better to spread out the number of TSM client over a period of time by using the randomizing feature of scheduling. By scheduling backups appropriately, server performance may improve.

If possible, schedule server processes such as expiration, storage pool backup, move data, export/import, etc, when client backups are not active.

LAN Free Backup

- Backup/restore to tape or disk using the SAN
- Meta data is sent to the server using the LAN
- Frees TSM server from handling data --> better scalability
- Potentially faster than LAN backup/restore
- Great for large file workloads, databases (TDPs)
- Small file workloads have bottlenecks other than data movement
- Requires the TSM Storage Agent on the client

- For LAN-free using disk you must implement Tivoli SANergy
- Ensure sufficient data paths to tape drives.
- Do not use if you bundle many (>20) separate dsmc commands in a script.
    - dsmc start/stop overhead higher due to tape mounts.
    - Use the new file list feature to backup a list of files.

TSM Storage Agent Tuning

- TSM Storage Agent has its own configuration file, dsmsta.opt, containing many of the same options as the server dsmserv.opt.
- Use the dsmsta setstorageserver command for some options
- In general use the same settings as recommended for the server.
- Ensure TCPNODELAY is YES (the default) on the Storage Agent and TSM server.
- On all platforms with LAN Free client, set LARGECOMmbuffers to NO.

  Note: The Storage Agent can not be installed on the same machine as the TSM server.

TDP for Domino for S/390

DFSMS HFS HIPER apars OW51210 and OW51732, and their resolving ptf's, have identified and fixed a performance problem with TDP for Domino, Domino for S/390 and HFS. Please ensure the resolving ptf's are installed on your system.

External TSM Tuning Tips

- NETWORKS
- AIX's VMTUNE
- JFS vs. Raw Logical Volumes
- Estimating Throughput
- EstimatingThroughput for Other Environments
- Tape Device Tuning Tips

NETWORKS

- Use dedicated networks for backup (LAN or SAN)
- Keep device drivers updated
- Ethernet speed and duplex settings - be sure they are the same for all connections on a given switch – DO NOT use autodetect settings
- Gb Ethernet jumbo frames (9000 bytes) - Only available if supported on client, server, and switch. Not all Gb Ethernet hardware supports jumbo frames. Jumbo frames can give improved throughput and lower host CPU usage especially for larger files.

AIX's VMTUNE

The RISC System/6000 Virtual Address space is managed by the Virtual Memory Manager (VMM). VMM is administered by the AIX command VMTUNE.

- The vmtune command is used to tune the AIX virtual memory system
- Provided in the bos.adt.samples fileset
- The vmtune options can be displayed using the 'vmtune' comand without any options.
- Options can be changed by specifying the appropriate option/value.
- Changes to vmtune parameters do not survive reboot. Settings are valid until the next reboot, so put in /etc/inittab.
- The vmtune command usually may be found in the following location: /usr/samples/kernel/vmtune

AIX - Read ahead (vmtune maxpgahead)

- When AIX detects sequential file reading is occuring, it can read ahead even though the application has not yet requested the data
- Improves sequential read performance on JFS filesystems
- TSM client - Improves large file backup throughput
- TSM server - Improves storage pool migration throughput
- Recommend setting: maxpgahead to maximum: -R 256
- When altering the read ahead parameter (-R) you must also alter the maxfree parameter (-F) so that there is enough free memory to store the read ahead data.
- The following equation must hold: minfree+maxpgahead<=maxfree
- To calculate minfree and maxfree use these formulae:
    - minfree = 120 multiplied by the number of processors
    - maxfree = 120 + maxpgahead (or j2_maxPageReadAhead) multiplied by the number of processors
- This does not improve read performance on raw logical volumes
- Using RLVs for the TSM server can cut CPU consumption but may be slower during storage pool migration. (Due to lack of read ahead)

Recommendation
-R 256 (maximum)

AIX File Cache (vmtune Minperm/Maxperm)

- Determines how much memory AIX sets aside for file system cache
- AIX can/will page out application (like TSM) memory in favor of caching filesystem data. This can cause paging of the database buffer pool leading to slow database performance.

- Paging of the database buffer pool can cause database cache hit statistics to be overly optimistic.
- TSM server does not benefit greatly from filesystem cache.
- Lowering maxperm will make AIX retain more application memory.
- Stop TSM server virtual memory paging by modifying the minperm/maxperm parameters. Exception: RAM constrained systems. Exception: database bufferpool size is too large. (see previous BUFPOOLSIZE tuning section for proper settings.)
- Good starting poing is setting aside a max of 50% (-P 50) for filesystem caching (instead of the default of 80%). Lower further if the changes was not effective (change realtime). As maxperm approaches minperm consider lowering minperm as well. Watch vmstat for progress, if pageouts go to zero pageins's will eventually lower as well.If the client and server are on the same processor use the vmtune parameter *vmtune -R 256 -F376 -c 1*.

JFS vs. Raw Logical Volumes

If RLV's are used for disk storage volumes on the TSM server, performance may be better than when JFS is used. When RLV's are used, the server does not go through AIX VMM system. This may result is lower cpu utilization for each disk I/O. On the other hand when reading from server disk volume, RLV will not use read ahead mechanism. This may result in poorer performance on restores and server move operations from disk to tape.

In general, we have seen a performance improvement when the TSM Recovery Log or TSM Data Base are put on RLV.

If you choose to use RLV instead of JFS, please heed the following warnings:

- Do not use AIX to mirror Raw Logical Volumes with TSM. Instead use TSM mirroring facilities. AIX puts information in a control block on disk which will overwrite TSM's control information.
- Do not change the size of RLV's using SMIT. Please refer to TSM for AIX Administrators Reference manual for EXTend DB and EXTend LOG commands.
- The TSM Administrator must be careful not to start multiple instances of TSM Server code which may use the same RLV's. With AIX there is no locking when using RLV. TSM attempts to fix this by implementing locks using files in the /tmp directory. These locks may be deleted if programs are run which clean up the files in the /tmp directory. For example, the Skulker program should not be used.
- Do not use raw logical volumes for HACMP implementations that include the TSM server. During an HACMP failover event, the takeover system can potentially access raw volumes before the failing system has completely released them

Solaris direct I/O should be used if not using raw volumes. For AIX, if you choose not to use raw logical volumes, direct I/O should always be used for jfs2 file systems. For jfs

file systems, direct I/O may cause degradation, especially with large-file enabled file systems.

Estimating Throughput

The TSM performance lab has conducted performance evaluations for some combinations of TSM server and client platforms and with various processor, disk and tape combinations. The results of these evaluations are not generally available to the public. However, results can be provided for specific situations. Contact your IBM representative if you are interested in specific results.

It's easy to estimate the throughput for workloads with average file sizes different from those that were tested in the TSM performance lab. However, the overall TSM environment must conform to one of the environments in one of our evaluation reports.

The first step is to find the table in an evaluation report for the TSM function and environment that matches your specific requirements.

The next step is to determine the average file size of the client workload for which the estimate is to be made. The following formulas call this value *EstFileSize* and expect this value in KB.

Then apply one of the following rules:

1. If the average file size is greater than 256 MB, then use the throughput in KB/second or GB/hour for the 256 MB workload.
2. If the average file size is less than 1 KB, then use the throughput in KB/second for the 1 KB workload. Throughput is effectively limited by the number of files that can be processed in a given amount of time. Calculate the throughput for the estimated file size in KB/sec using the following formula:

   *Throughput(KB/sec) = 1KBThroughput(KB/sec) * EstFileSize(KB)*

   Convert KB/sec to GB/hr using the following formula:

   Throughput(GB/hr) = Throughput(KB/sec) * 3600 / 1048576

3. If the average file size is greater than 1 KB and less than 256 MB, then calculate the throughput using the two known measurement points in the table that most closely bound the estimate point.

   A. Obtain the following values from the table for the function and environment of interest:

   LowerFileSize - average file size in KB at the lower measurement point
   LowerThroughput - throughput in KB/s at the lower measurement point

UpperFileSize - average file size in KB at the upper measurement point
UpperThroughput - throughput in KB/s at the upper measurement point

B. Calculate the throughput in KB/sec using the following formulas:

A = **log**(*UpperThroughput(KB/sec)*/*LowerThroughput(KB/sec)*)
B = **log**(*EstFileSize(KB)*/*LowerFileSize(KB)*)
C = **log**(*UpperFileSize(KB)*/*LowerFileSize(KB)*)
D = **log**(*LowerThroughput(KB/sec)*)
Throughput(KB/sec)= 10 ** ((*A* * *B* / *C*) + *D*)

All references to **log** in the formulas shown above imply log base 10.

Estimating Throughput for Other Environments

Estimating throughput for environments that have not been directly tested can be more difficult. However, the following important observations can be made:

- Throughput over a network can be expected to reach saturation at around 80 percent of its rated capacity. Efficiency indicates the percentage of maximum throughput rate that can realistically be achieved. This leads to the following maximum throughputs that can be obtained for given networks:

Network	Mb/sec	MB/sec	GB/hr	% Efficiency
Ethernet	10	1.0	5.7	40
Token Ring	16	1.6	5.7	80
Ethernet	100	10.0	17.6	40
FDDI	100	10	35.2	80
ATM	155	15.5	34.1	50
SPSwitch		120	264	50
T3	45	4.48	15.8	80
T1	1.54	.16	.56	80
Gigibit Ethernet	1GB	100	tbd	80

- Throughput for backup and restore of small file workloads is basically independent of network type, as long as the network remains unsaturated, and propagation delays are not excessive due to intervening routers or switches.

Tape Device Tuning Tips

Configure enough tape drives for:

- The maximum number of TSM client sessions backing up direct to tape at any time during the peak backup window
- Add additional tape drives for other functions that run during the backup window: Storage pool migration, storage pool backup, reclamation.

Tape Drive Cleaning:

- Cleaning the tape drive per manufacturer's specifications is very important to ensure maximum tape drive performance. Failure to clean tape drives can cause read/write errors, drive failures and generally poor performance at the tape drive.

Tape Drive Compaction:

- Use the DEFINE DEVCLASS command and modify the FORMAT parameter to enable tape compaction. In many cases, enabling compaction at the tape drive will improve TSM throughput. To enable the tape drive to use compaction, set the appropriate recording format at the tape drive. The FORMAT option of the DEFINE DEVCLASS command specifies the recording format to be used when writing data to sequential access media. The default is DRIVE, which specifies that TSM selects the highest format that can be supported by the sequential access drive on which a volume is mounted, which usually allows the tape control unit to perform compaction.

Warning
Avoid specifying the DRIVE value when a mixture of devices are used within the same library. For example, if you have drives that support recording formats superior to other drives in a library, do not specify the FORMAT=DRIVE option. Please refer to the appropriate TSM Administrators Guide for more information.

If you do not use compression at the client and your data is compressible, you should achieve higher system throughput if you use a compaction at the tape control unit. Refer to the appropriate TSM Administrators Guide for more information concerning your specific tape drive. If you compress the data at the client, we recommend that you not use compaction at the tape drive. In this case, you may lose up to 10-12% of the tape capacity at the tape media.

Chapter 4. Storage Device Hints and Tips

IBM LTO Drives

There are many factors that affect a tape drive's sustained transfer rate including:

- Native transfer rate
- Compression ratio
- Blocksize

- File size
- Server attachment
- Server attachment HBA type
- Disk transfer rate
- Network bandwidth
- Server utilization
- Start/stop performance
- Application control file activity
- Bus bandwidth
- Quality of Media

The sustained transfer rate takes into account the net effect of all these factors.

Performance Considerations

The IBM LTO Ultrium tape drive has a native streaming data rate of up to 15MBs/sec, and up to 30MBs/sec with 2:1 compression. Streaming rate is the rate at which a tape drive can read/write, not including any start/stop operation. Most uses of tape do include some start/stop which slows down the sustained rate at which the drive operates.

For example, when writing to a tape drive, normally the drive returns control to the application when the data is in the tape drive's buffer, but before the data has actually been written to tape. This mode of operation provides all tape drives a significant performance improvement. However, the drive's buffer is volatile. If the application wants to be absolutely sure the write makes it to tape, the application needs to flush the buffer. Flushing the buffer causes the tape drive to backhitch (start/stop). The TSM parameters TXNBytelimit and TXNGroupmax control how frequently TSM issues this buffer flush command.

When writing to a tape drive, network bandwidth must be considered. For example, 100BaseT ethernet LAN can sustain 5 to 6 MBs/sec. Therefore, obviously, you will not be able to backup to LTO (or any tape drive) any faster than that.

Performance Recommendations

When using LTO/ULTRIUM drives with TSM it is important to use the following settings to enhance performance.

TSM Server:

- TXNGroupmax 256
- MOVESizethresh 2048
- MOVEBatchsize 1000

TSM Client:

- TXNBytelimit 2097152

If TSM clients have, on average, files smaller than 100KB it is recommended that these clients backup to a disk storage pool for later migration to tape. This allows more efficient data movement to tape.

When IBM 3580 LTO drives are used make sure that the drive microcode is at the most current level. Instructions for verifying the current LTO drive microcode release and how to install the new release can be found in the IBM Ultrium Device Drivers Installation and Users Guide. This manual can be found at url:

- Ftp://ftp.software.ibm.com/storage/devdrvr/

Reference Material

- Using IBM LTO Ultrium with Open Systems (SG24-6502-00 IBM Redbook Redpiece
- Implementing IBM LTO Tape in Linux and Windows (SG24-6268 IBM Redbook)
- Designing an IBM Storage Area Network, SG24-5758
- Planning and Implementing an IBM SAN, SG24-6116
- IBM SAN Survival Guide, SG24-6143
- Using Tivoli Storage Manager in a SAN Environment, SG24-6132
- The IBM LTO Ultrium Tape Libraries Guide, SG24-5946

Other resources

These publications are also relevant as further information sources:

- 3584 UltraScalable Tape Library Planning and Operator Guide, GA32-0408
- 3583 Ultrium Scalable Tape Library Setup and Operator Guide, GA32-0411
- 3581 Ultrium Tape Autoloader Setup, Operator and Service Guide, GA32-0412
- 3580 Ultrium Tape Drive Setup, Operator and Service Guide, GA32-0415
- IBM Ultrium Device Drivers Installation and User’s Guide, GA32-0430
- IBM Storage Area Network Data Gateway Module Setup, Operator and Service Guide, GA32-0436
- TSM for AIX Managed System for SAN Storage Agent User’s Guide GC36-0001
- TSM for Sun Solaris Managed System for SAN Storage Agent User's Guide, GC36-0002

SSA Disks

SSA controllers with cache do indeed make RAID-5 more attractive than the non-cached. However, there is still a performance penalty with writes. For the size of writes that the TSM DB is concerned, the cached controllers may give somewhere in the order of 2x performance over their non-cached counterparts. These (cached controllers) may give

about 5-10 MB/s. The cached controllers can achieve about 50% of a non RAID-5 configuration.

This write penalty happens because RAID-5 needs to do four I/O's per write request (versus one I/O per write request in a non RAID-5/mirrored environment).

As for TSM performance measurements and recommendations with SSA, at this time, there are none. Please consider the trade-off's between performance and costs/disk space. The question becomes, how important is the TSM DB I/O? Some will say that it's not a big deal, so RAID-5 is a reasonable consideration. It comes down to finding the throughput bottleneck. In many cases the bottleneck is found to be the disk I/O sub-system. In these cases, RAID-5 is probably not a good choice.

SilverNode SCSI Slot Positioning

We have found that the placement of SCSI adapters on the H50 is very critical. A document dealing with slot placement for best/better performance (PCI-based systems) is available on the RS/6000 Home page at: http://www.rs6000.ibm.com/ resource/hardware_docs/,"PCI Adapter Placement Reference, SA38-0538-04."

Unfortunately, this document does not cover the 332MHz SMP Node (Silver) specifically, even though it is very similar to the F50 (Wildcat) and H50 (Silver) systems.

In field tests of a 332MHz SMP Node (Silver node), Ethernet cards were removed from the slots in the wide node expansion chassis labeled I6, I7, I8 (I5 is also on the same PCI controller as the last 3 slots). The AIX location codes for these slots were:

- I5 (2F-00 through 2F-07)
- I6 (2F-08 through 2F-0F)
- I7 (2F-10 through 2F-17)
- I8 (2F-18 through 2F-1F)

There are some adapter rules listed in the announcement letter for the 332MHz SMP Nodes, ALET 198-086. The FC 6215 adapter is specifically listed as unable to reside in the last 4 slots, PCI SSA Multi-Initiator/RAID EL Adapter.

Summary:

To check r/spoolsize use: lsattr -E -l css0

To reset r/spoolsize use: (from directory /usr/lpp/ssp/css)

- chgcss -l css0 -a rpoolsize=in_bytes
- chgcss -l css0 -a spoolsize=in_bytes

If r/spoolsize < 4M, try setting to 4M

Note: You'll have to reboot the node for the chgcss changes to take effect.

Busses

- If your machine has multiple PCI busses, spread out high-throughput adaptors among the different busses. For example if you are going to do a lot of backups to disk you probably don't want your network card and disk adaptor on the same PCI bus.
- Theoretical limits of busses are just that, theoretical. Though you should be able to get close in most cases.
- As a general rule it is best to have only one or two tape drives per SCSI bus and one to four adapters per fiber HBA

Appendix A. Using TSM's Archive Function

Abstract

Tivoli Storage Manager (TSM) has offered an archive function since ADSM Version 1. The archive function has, however, undergone functional changes as TSM has progressed through several versions. The archive function is a powerful way to store inactive data with a fixed retention time. However, some TSM users have been frustrated when using the archive function because of inadequate performance or a perceived lack of functionality. This paper will review the functional changes that the archive function has undergone plus provide some implementation advice for those TSM users who plan to use TSM archives.

History

The TSM client provides two basic functions: Backup/Restore and Archive/Retrieve. The design intent for Backup/Restore is to provide protection against data loss. The Backup/Restore function includes versioning and management of the active set of data for full system restore. The design intent for Archive/Retrieve is to house inactive data for some fixed retention period. This could be individual files or groups of files. Archive will work also for large-scale archive of an entire system. In ADSM Version 1 and Version 2, archive was focused purely on files—directories were not eligible for archive. ADSM Version 1 would rebuild directories on retrieve, if necessary, but they were rebuilt with default properties. An archive could assign a “Description” to a file as it is archived but the “Description” had no further significance and the default “Description” was the null string.

Archive functionality changed significantly in ADSM Version 3.1. The ADSM Client Graphical User Interface (GUI) introduced a nested presentation that displays: 1) All

Descriptions 2) All filesystems within a Description 3) All directories within a filesystem and 4) All files within a directory. This gave new significance to the archive Description—it was now the name of an “Archive Package”. Also, the default Description was changed to be Date and Time stamp. To improve TSM database performance, new internal tables were added for indexing by Description. These tables were populated when a TSM client first used the Version 3.1 GUI interface to perform an Archive. This process is known as an “Archive Conversion”. To further improve the overall functionality, archive of directories was added. To ensure that a directory would not be expired before all of its archived file entries were expired, directories were, by rule, assigned the Management Class with the longest retention. These changes were also in effect for the TSM Client Command Line Interface (CLI).

These changes in ADSM Version 3.1 gave needed functions to archive but they created problems in some specific circumstances. If clients repeatedly archived the same set of files or used command file driven archives that repeatedly invoked the CLI, they could see explosive growth of the TSM Database and many duplicates of archived directories. The problem was further compounded by the fact that directories were assigned to the Management Class with the longest retention. In some cases, where an Archive Copy Group with retention of NO LIMIT was available, directory archives would never expire.

At the Version 3.1.0.7 level, client code was added to ensure that only unique directories were archived. As part of archiving a file, the client first queries the server to see if a directory archive exists (Description was included as part of the unique directory identification). Also, the timestamp was removed from the default Description. Several changes were made on the TSM server at this code level. Utilities were created to clean up the excess directory archives. Inventory Expiration was changed such that it would check to see if an archived directory was referenced before that directory would be expired. Since this ensured that no directory would be expired before its files were expired, the directory archives were now assigned to the Management Class with the shortest retention. Unfortunately, most pre-Version 3.1.0.7 directory archives included the timestamp so the utilities were not effective. The other changes had the effect of slowing the Archives and Inventory Expirations (especially where there already were many directory archives).

With Version 4.2.1, a change was made such that when the Client queries the Server for a directory, if it exists, the archive date on the directory is changed to the archive date of the archived file that most recently referenced it. This delays when the directory qualifies for expiration and thus reduces the number of “is it referenced” tests at Inventory Expiration time. A new internal table was added to the TSM database to improve search time for large numbers of archives of files with the same Description. Also, a new utility, CONVERT ARCHIVE was added which allows the TSM Administrator to control when a node is converted to use the new internal tables in the TSM database. Additionally, at this level, a new client option was added, “-v2archive”, which, when specified, would prevent directories from being archived—essentially reverting to the Version 2 functionality. These functions were also built into the TSM Version 3.7.4 level.

Implementing Archive

The first step in implementing archive should be to determine if archive is indeed the proper function for the business problem. Archive of a set of business related files, even on a repeated basis, is appropriate. Also, use of archive to prepare a “snapshot” of a set of files or directories before making a significant change is acceptable. Some archive practices that should be avoided are:

- Attempts to implement a point-in-time recovery strategy: this usually means frequent, large-scale archives. These kinds of archives tend to exacerbate the problems described above by flooding the TSM Database with many directory and file entries. Often the requirement can be satisfied with TSM client backup function or with TSM Server generated Backupsets.
- Replacement for a tape rotation strategy: in some cases there is a need to implement, with TSM, a tape management strategy that regularly and predictably rotates specific tapes off-site and then, at a predefined time, back on-site. This is not TSM’s normal mode of operation but can be accomplished various ways. Using Archive to accomplish tape rotation is also likely to create problems with excessive directory and file entries in the TSM Database. A more efficient way to do this by using the client Backup function and a series of Storage Pools. Backups are directed to the Storage Pools in a round-robin fashion. When it is time to recover the tape volumes from the off-site location, all the volumes are deleted from that Storage Pool.

Whenever possible, Backupsets should be used as a replacement for archives. Backupsets do not generate any of the problems described above and Backupsets have a much smaller impact on the TSM Database—each Backupset creates only one entry in the TSM Database. If Backupsets are not appropriate then another option might be to aggregate the files and directories before archiving them (i.e. PKZIP, tar, etc.).

Symptoms

These are some of the symptoms that may indicate an archive induced problem:

- Slow Archive or Retrieve throughput: Archive performance will degrade as more and more duplicate directory entries are Archived Archive throughput may degrade over a period of days or months and/or Archive throughput may get gradually worse as the archive progresses—it may even get so slow that it looks as if the archive has stopped. Be sure to examine all other tuning variables first to be sure the problem is not one that can be fixed by parameter changes. Use the TSM Performance Tuning Guide which is available at: http://www.tivoli.com/support/public/Prodman/public_manuals/storage_mgr /v3pubs/pfguide.htm.
- TSM Server Inventory Expiration duration: Inventory Expiration may take longer and longer because it must process an increasing load of duplicate Archived

directories. Also, for each expired directory archive, Inventory Expiration must check for dependant files. This tends to compound the problem when there are many duplicate Archived directories.

- Excessive Growth of the TSM Database: The TSM Database grows because each Archived directory requires an entry in the TSM Database. Be sure to consider other possible causes of TSM Database growth: New clients; changes in Management Class retention or versioning; or heavy growth from existing clients.

Suggested Actions

If you are experiencing any of the symptoms described above and you think it might be related to excessive Archived directories, the best action to take is to contact the IBM/Tivoli Software Support Center. They will help you use the service utilities to examine your system and to take corrective action if it is appropriate. You may be asked to run these commands:

Q OCC TYPE=ARCHIVE

CLEANUP ARCHDIR SHOWSTATS

The duplicates value indicated in the CLEANUP ARCHDIR output is based on duplicates by path and directory name only. What you are looking for in this output is a large number of duplicates—hundreds of thousands at least—or a high ratio of directories to files. These may indicate a problem.

NOTE: DO NOT USE THE CLEANUP ARCHDIR UTILITY UNLESS DIRECTED TO DO SO BY AN IBM OR TIVOLI SUPPORT REPRESENTATIVE.

If it is determined that you are experiencing a problem with excessive directory Archives you may be directed to one or more of these solutions:

- Upgrade your TSM Server to Version 4.1.2 or beyond. This level introduces new options and utilities to deal with the excessive directory archives.
- Use the CLEANUP ARCHDIR to eliminate some or most of the unnecessary directory Archives. This utility is not effective in all cases depending on when and how you arrived at the current condition.
- Unconvert the Node and then Convert the Node. This will force the node to use the new table structure within the TSM Database. This new structure is more efficient.
- Use the –V2ARCHIVE option when you create the Archives. This option reverts the Archive process to Version 2 functionality. This means that directories are not Archived. On Retrieve, TSM will, if necessary, create directories with default properties. Use this option if you can tolerate Archive of files only.
- Use a blank (not null) character string for the Archive Descriptions. This will prevent excessive “unique” Archives of the same path and directory name.

NOTE: The CLEANUP ARCHDIR command has been changed to UPDATE ARCHIVE in ITSM 5.2.2

Summary

Archive functionality has changed over that last several product versions and releases. Understanding these changes can help you to diagnose your problem.

Signs of excessive directory archives are: slow performing Archives or Retrieves; long running Inventory Expiration; and/or rapid growth of the TSM Database.

If you are having problems with Archives, you should explore alternatives such as Backupsets.

Contact IBM or Tivoli Software support for help and advice if you think you have a problem with excessive directory Archives.

Appendix B. Network Protocol Tuning

TCP/IP Tuning Considerations

Communication Concepts

This chapter will focus on tuning the TCP/IP protocol to obtain maximum TSM performance. We shall very briefly discuss the underlying concepts involved in TCP/IP before proceeding to the tuning aspects.

To tune TCP/IP, the most significant performance improvements are found by modifying parameters that affect the following key areas:

- Data transfer block size
- Window values
- Connection availability

The following tasks require system resources:

- Keeping communication connections available
- Keeping user data until it is acknowledged (on the transmit side)
- Managing the communications layers.

These resources include memory, CPU, communications adapters, link utilizations, and involve the limitations of various communication layer implementations. Data sizes and

flow control are the two main factors that cause resource over-commitment, which results in system performance degradation.

TCP/IP Protocol and Functions

The TCP/IP protocols and functions can be categorized by their functional groups: the network layer, internetwork layer, transport layer, and application layer. Table 2 shows the functional groups and their related protocols.

Table 2. TCP/IP Functional Groups

Group	Protocols and Functions
Network layer	Token-Ring Ethernet Others
Internetwork Layer	Internet Protocol (IP) Internet Control Message Protocol (ICMP) Address Resolution Protocol (ARP)
Transport Layer	Transmission Control Protocol (TCP) User Datagram Protocol (UDP)
Application Layer	Telnet File Transfer Protocol (FTP) Remote Procedure Call (RPC) Socket Interfaces Others

Protocol Functions

The protocol functions can be categorized as the following:

- Reliable delivery
- Message assembly and disassembly
- Connection control
- Flow control
- Error handling

Reliable Delivery

Reliable delivery services guarantee to deliver a stream of data sent from one machine to another without duplication or loss of data. The reliable protocols use a technique called acknowledgment with retransmission, which requires the recipient to communicate with the source, sending back an acknowledgment after it receives data.

Assembly and Disassembly of Packets

Each layer of a communications protocol can potentially perform some sort of Assembly or Disassembly function. If the source and destination nodes do not lie on the same physical network, then the TCP/IP software has to fragment the packets traveling from one network to another if the Maximum Transmission Units (MTUs) on the networks do not match. The TCP/IP software at the receiving station then reassembles the fragments.

There are a number of advantages to Assembly/Disassembly:

- A communications network may only accept data blocks up to a certain size, hence requiring that larger blocks be broken down. For example, an Ethernet LAN has an MTU size of 1500 bytes, whereas a Token-Ring LAN has an MTU size of up to 16000 bytes.
- Error control may be more efficient for smaller blocks.
- More equitable access, with shorter delay, may be provided to shared transmission facilities. For example, if the line is slow, allowing too big a block to be transmitted could cause a monopolization of the line.

Disadvantages of Assembly/Disassembly are:

- Each transmitted unit of data requires some fixed amount of overhead. Hence the smaller the block, the larger the percentage of overhead
- More blocks have to be processed for both sending and receiving sides in order to transmit equal amounts of user data, which can take more time.

Flow Control

Flow Control is a function provided by a receiving system that limits the amount or rate of data that is sent by a transmitting system. The aim is to regulate traffic to avoid exceeding the receiver's system resources.

Error Control

Error Control is needed to guard against loss or damage of data and control information. Most techniques involve error detection and retransmission.

Sliding Window

The need to use communication channels efficiently, both in terms of Flow Control and Error Control, gave rise to the concept of the Sliding Window. To achieve reliability, the sender sends a packet and waits until it gets an acknowledgment before transmitting another. The Sliding Window protocol enables the sender to transmit multiple packets before waiting for an acknowledgment. The advantages are:

- Simultaneous communication in both directions
- Better utilization of network bandwidth, especially when there are large transmission delays.

- Traffic flow with reverse traffic data, known as piggybacking. This reverse traffic may or may not have anything to with the acknowledgment that is riding on it.
- Variable window size over time. Each acknowledgment specifies how many octets have been received and contains a window advertisement that specifies how many additional octets of data the receiver is prepared to accept, that is, the receiver's current buffer size. In response to decreasing window size, the sender decreases the size of its window. Advantages of using variable window sizes are flow control and reliable transfers.

A client continually shrinking its window size is an indication that the client cannot handle the load, and thus increasing the window size does not improve performance.

TCPWindowsize (TSM option)

TCPBuffsize (TSM option)

TCPNodelay (TSM option)

Platform Specific Recommendations

AIX Servers and Clients

It is important to minimize all performance constraints on AIX to achieve maximum throughput on the server. This is accomplished by tuning the network option parameters on AIX.

TSM uses TCP/IP communication protocol over the network. It is important to tune the TCP protocols to obtain maximum throughput. This requires changing the network parameters that control the behavior of TCP/IP protocols and the system in general.

In AIX, an application using the TCP/IP communication protocol opens a TCP socket and writes data to this socket. The data is copied from the user space into the socket send buffer, called the tcp_sendspace in kernel space. The receive buffers are called tcp_recvspace. The send and receive buffers are made up of smaller buffers called mbufs.

An mbuf is a kernel buffer that uses pinned memory and comes in two sizes, 256 bytes and 4096 bytes called mbuf clusters or simply clusters. The maximum socket buffer size limit is determined by the sb_max kernel variable. Be cause mbufs are primarily used to store data for incoming and outgoing network traffic, they must be configured to have a positive effect on network performance. To enable efficient mbuf allocation at all times, a minimum number of mbuf buffers are always kept in the free bufferpool. The minimum number of mbufs is determined by lowmbuf, whereas the minimum number of clusters is determined by the lowclust option. The mb_cl_hiwat option controls the maximum number of free buffers the cluster pool can contain.

The thewall network option controls the maximum RAM that can be allocated from the Virtual Memory Manager (VMM) to the mbuf management routines. The netstat -m can be used to obtain detailed information on the mbufs. The netstat - I tr0 command can be used to determine if there are errors in packet transmissions.

If the number is greater than 0, overflows have occurred. At the device driver layer, the mbuf chain containing the data is put on the transmit queue, and the adapter is signaled to start the transmission operation. On the receive side, packets are received by the adapter and then are queued on the driver-managed receive queue. The adapter transmit and receive queue sizes can be configured using the System Management Interface Tool (SMIT).

At the device driver layer, both the transmit and receive queues are configurable. It is possible to overrun these queues. To determine this use netstat -v command, which shows Max Transmits Queued and Max receives Queued.

Table 3. Transmit Adapter Queue Settings

Bus Type	Queue Setting
PCI	256
Micro-Channel	512-2048

Table 4. MTU and MSS Settings

Network Type	MTU	MSS (RFC1323 0)	MSS (RFC1323 1)
FDDI	4352	4312	4300
Token Ring	4096	4056	4044
Ethernet	1500	1460	1448

The MTU is another important factor. For best throughput for systems on the same type of network, it is advisable to use a large MTU. In multi-network environments, if data travels from a network with a large MTU to a smaller MTU, the IP layer has to fragment the packet into smaller packets (to facilitate transmission on a smaller MTU network), which costs the receiving system CPU time to reassemble the fragment packets. When the data travels to a remote network, TCP in AIX defaults to a Maximum Segment Size (MSS) of 512 bytes. This conservative value is based on a requirement that all IP routers support an MTU of at least 576 bytes. The default MSS can be overridden in the following three ways:

1. Specify a static route to a specific remote network and use the -mtu option of the route command to specify the MTU to that network. Disadvantages of this approach are:
    - It does not work with dynamic routing.
    - Impractical when the number of remote networks increases.

    - Routes must be set at both ends to negotiate a value larger than a default MSS.
2. Use the tcp_mssdflt option of the no command to change the default value of MSS. This is a system wide change. In a multi-network environment with multiple MTUs, the value specified to override the MSS default should be the minimum MTU value (of all specified MTUs) less 40. In an environment with a large default MTU, this approach has the advantage that MSS does not need to be set on a per-network basis. The disadvantages are:
    - Increasing the default can lead to IP router fragmentation if the destination is on a remote network, and the MTUs of intervening networks is not known.
    - The tcp_mssdflt parameter must be set to the same value on the destination host.
3. Subnet and set the subnetsarelocal option of the no command. Several physical networks can be made to share the same network number by subnetting. The subnetsarelocal option specifies, on a system-wide basis, whether subnets are to be considered local or remote networks. With subnetsarelocal=1 (the default), Host A on subnet 1 considers Host B on subnet 2 to be on the same physical network. The consequence of this is that when Host A and Host B establish connection , they negotiate the MSS assuming they are on the same network.

    This approach has the following advantages:

    - It does not require any static bindings MSS is automatically negotiated.
    - It does not disable or override the TCP MSS negotiation so that small differences in the MTU between adjacent subnets can be handled appropriately.

    The disadvantages are:

    - Potential IP router fragmentation when two high-MTU networks are linked through a lower-MTU network.
    - Source and destination networks must both consider subnets to be local.

Please refer to the AIX Performance Tuning Guide (SC23-2356) for details.

Recommendation
In an SP2 environment with a high speed switch, use MTU of 64KB.

AIX - no (network options)

The network option parameters can be configured using the no command.

- Use no -a to view current settings
- When using TCP window sizes > 64, set rfc1323 to 1

- If you see non-zero "No mbuf errors" in entstat, fddistat, or atmstat, raise thewall
- Recommend setting thewall to at least 131072 and sb_max to at least 1310720
    - Newer versions of AIX have larger defaults (don't lower)
- no settings do not survive reboot, so add to /etc/inittab. Recommended changes:
    - no -o rfc1323=1
    - no -o thewall=131072
    - no -o sb_max=1310720

Recommended values for the parameters described in this section areas follows:

Network options

`lowclust = 200`
`lowmbuf = 400`
`thewall = 131072`
`mb_cl_hiwat = 1200`
`sb_max = 1310720`
`rfc1323 = 1`

Note: The lowmbuf, lowclust and mb_cl_hiwat options are applicable only for AIX v3.2.x and not for AIX v4.1.x. In AIX v4.1.x,if setting sb_max to 1310720 results in an error message when running TCP/IP applications, then please set the sb_max value to 757760.

MVS Server with IBM TCP/IP for MVS

This section describes how to configure the TCP/IP address space for IBM TCP/IP for MVS. During initialization of the TCP/IP address space, system operation and configuration parameters are read from a configuration dataset. The program searches for the data set job_name.node_name.TCPIP, where node is the node name of the system as specified on the VMCF initialization record. VMCF is a subsystem defined by a line in the IEFSSNxx member that causes the VMCF address space to be created and initialized. If this dataset is not found, the program uses the first of the following data sets it finds:

- tcpip.node_name.TCPIP
- job_name.PROFILE.TCPI
- tcpip.PROFILE.TCPIP

We only discuss the configuration parameters that affect overall system performance. The various free pool sizes can be configured depending on the user environment and are discussed below. In our lab environment, default values were used, except as noted. These settings serve as our recommended values. However, these values may need to be altered depending on system capacity requirements.

TCPIP.DATA

TCPIP.DATA contains hostname, domainorigin, nsinteraddr (name server), etc. The content of TCPIP.DATA is the same as for previous releases of TCP/IP for

MVS. For a sample TCPIP.DATA, please see the IP Configuration manual (SC31-8513) or reference the sample shipped with the product. One important recommendation is to keep the statement "TRACE RESOLVER" commented out to avoid complete tracing of all name server queries. This trace should be used for debugging purposes only.

PROFILE.TCPIP

During initialization of the TCPIP stack, configuration parameters for the stack are read from the PROFILE.TCPIP configuration data set. Reference the OS/390 IP Configuration manual (SC31-8513) for additional information on the parameters that are used in this file.

The PROFILE.TCPIP contains TCP buffer sizes, LAN controller definitions, server ports, home IP addresses, gateway statements, VTAM LU's for Telnet use, etc.

TCP/IP in CS OS/390 V2R5 and later releases have been simplified by removing TCP/IP buffer pool and control block definitions from PROFILE.TCPIP, i.e., no tuning is needed for TCP/IP buffers in the PROFILE.TCPIP; buffers are dynamically allocated by communication storage manager (CSM).

TCP SEND/RECV BUFFER SIZES ----------

When send/recv buffer size(s) are not specified in the PROFILE, a default size of 16K will be used for send/recv buffers and a default of 32K will be used for the TCP window size. If send/recv buffer size(s) are specified, they will be used as specified and the TCP window size will be set to twice the TCP recv buffer size up to a maximum of 65535. .

Note:RFC 1323 which supports larger than 64K-1 TCP window size is available in V2R7. Customers can specify the send/recv buffer size(s) on the TCPCONFIG statement:

```
TCPCONFIG     TCPSENDBFRSIZE    64K
              TCPRCVBUFRSIZE    64K

              TCPSENDBFRSIZE    256K (OS/390 server using Jumbo Frame
              TCPRCVBUFRSIZE    256K  Gigabit Ethernet)

              TCPSENDBFRSIZE    64K-128K (OS/390 server using ATM)
              TCPRCVBUFRSIZE    64K-128K
```

Socket applications can override these values for a specific socket by using the setsockopt call:

- setsockopt(SO_SNDBUF) or
- setsockopt(SO_RCVBUF) in socket application.

Important note for FTP Server and Client: The FTP server and client application override the default settings and use 64K-1 as the TCP window size and 180K bytes for

send/recv buffers. Therefore there is no change required in the TCPCONFIG statement for FTP server and client.

The DATABUFFERPOOLSIZE is no longer used in TCP/IP on OS/390. The new parameters TCPSENDBFRSIZE and TCPRCVBUFRSIZE do need to be set on OS/390 as well as the TCPWIndowsize on the client.

GATEWAY Statement

The GATEWAY statement indicates how to route datagrams to the specified network or host.

GATEWAY
`GATEWAYnetwork first_hop link_name max_packet_size subnet_mask subnet_value9.0.0 = PCN1 4000 0.255.255.0 0.113.1.0`

The parameter in the GATEWAY statement that affects performance is the max_packet_size, which is discussed in detail. The other parameters are discussed briefly.

network
: The internet address in dotted decimal form for the network.

host
: The host address specified as four octets.

first_hop
: Is either an (=) implying that messages are routed directly to destination on that network (or directly to that host) or is the internet address of a gateway or router.

link_name
: The name of the link through which packets are sent to the specified network. The link name is defined in LINK statement.

max_packet_size
: The maximum transmission unit in bytes for the network or host. The default is 576 bytes.

subnet_mask
: A bit mask expressed in dotted-decimal form that shows the bits of the host field that make up the subnet field.

subnet_value
: The value of the subnet field. It is expressed in dotted decimal notation.

Maximum transmission unit size considerations:

- The maximum packet size that the TCP/IP for MVS can handle is the large envelope size. Some networks limit the packet size to a smaller value.
- Information is transferred over a TCP connection in discrete packets. Each packet includes a TCP header and an IP header. The header size is independent of the amount of user information included, so the larger the packets sent, the less

relative bandwidth is consumed by protocol headers. Also, the TCP software consumes a fixed amount of CPU time for each packet, independent of the packet size, so the larger the packets sent, the less CPU is consumed.

- Large packets can be fragmented, each with its own headers by intervening gateways. Fragmentation and re-assembly of packets are expensive in their use of bandwidth and in CPU time. Therefore, packets sent through gateways to other networks should use the default size unless all intervening gateways and networks are known to accept larger packets.

PACKET SIZE on GATEWAY and BSDROUTINGPARMS

Make sure that you specify a packet size explicitly instead of using the word DEFAULTSIZE. The word DEFAULTSIZE requests that TCPIP supply a default value of 576 bytes, which might not be optimal in your configuration.

We recommend using the following sizes instead of DEFAULTSIZE as the packet size for the specified networks:

- 1492 bytes for Ethernet 802.3
- 1500 bytes for Ethernet Version 2 IEEE
- 1500, or 2000 or 4000 bytes (or greater if your environment permits) for token ring
- 4000 or 2000 bytes for FDDI (use larger value as permitted by your environment)
- 65527 bytes for CTC
- 4096 bytes for CLAW

Example:

```
GATEWAY
;                                Packet  Subnet        Subnet
; Network    First hop  Driver  size    mask          value
  9         9.67.118.43 FDDI1  4000    255.255.255.0 9.67.115.0
  DEFAULTNET 9.67.115.1  FDDI1  4000    0
```

or, if you use dynamic routing:

```
BSDROUTINGPARMS TRUE
;                              Subnet        Subnet
; Driver  Packet size  metric  mask          value
  FDDI1   4000         0       255.255.255.0 9.67.115.0
ENDBSDROUTINGPARMS
```

Please note the true/false option on the BSDROUTINGPARMS statement. If you specify FALSE, all IP packets that are sent to a destination that is more than one hop away will not use the packet size of the interface, but will use a default packet size of 576 bytes. If

you specify TRUE, all IP packets that are sent over an interface will use the interface's packet size, regardless of whether the destination is one or multiple hops away.

Recommendation
MTU = 4000 if no fragmentation occurs enroute (TR). MTU = 1500 if no fragmentation occurs enroute (EN).

TCP/IP and OS/390 Unix System Services Performance Tuning

- Make sure client/server TCP Window size is set to allowed maximum. Recommendation: Set the TCP window size on MVS to the allowed maximum by setting TCPRCVBUFRSIZE to 32K or larger and, if client workstation permits, use 65535 as the client window size. If the installation is storage constrained, however, use the default TCPRCVBUFRSIZE of 16K.
- Make sure client and server MTU/packet size are equal. Follow the recommendations given in the PROFILE.TCPIP section
- Make sure TCP/IP and all other traces are turned off for optimal performance. Trace activity does create an extra processing overhead
- Follow the OS/390 Unix System Services performance tuning guidelines in the OS/390 Unix System Services Planning manual (SC28-1890) or at this URL: http://www.s390.ibm.com/oe/bpxa1tun.html
- Region size(s) and dispatching priority: It is highly recommended to set the region size to 0K or 0M for the TCPIP stack address space and for started tasks such as the FTP server, the SMTP/NJE server, the Web server, the TSM server, etc.

  If your environment permits, set the dispatching priority for TCPIP and VTAM equivalent and keep servers slightly lower than TCPIP and VTAM.

  For other started tasks such as FTP, TSM: slightly lower than TCPIP task

  If you are using Work Load Manager, follow the recommendations above when your installation defines performance goals in a service policy. Service policies are defined through an ISPF application and they set goals for all types of MVS managed work.

- If you are using TCP/IP V3R2 reference "MVS TCP/IP V3R2 Performance Tuning Guide" (SC31-7188). This tuning guide also includes step by step process for tuning other TCP/IP platforms such as AIX, OS/2, DOS and VM.
- Update your PROFILE.TCPIP, TCPIP.DATA and FTP.DATA files with the applicable recommendations discussed in this APAR.
- Estimate how many OS/390 Unix System Services users, processes, ptys, sockets and threads would be needed for your OS/390 Unix installation. Update your BPXPRMxx member in SYS1.PARMLIB.
- Spread OS/390 Unix user HFS datasets over more DASD volumes for optimal performance.

- Monitor your OS/390 Unix resources with RMF and/or system commands (DISPLAY ACTIVE, and DISPLAY OMVS, etc.)

MVS Server with Interlink TCP/IP for MVS

This section describes how to configure Interlink SNS/TCPaccess for optimum TSM client/server performance. SNS/TCPaccess is a communication subsystem for Internet protocols that runs on MVS.

SNS/TCPaccess is implemented as an authorized subsystem with multiple task groups. The Application Control Protocol (ACP) task group supports the Internet application protocols and interfaces to the network hardware. The Application Program Interface (API) task group interfaces between the ACP transport protocol provider and transport protocol users in other address spaces. Other task groups may be required to build a complete system, but we will limit our discussion to these, as they are the ones most likely to influence TSM client/server performance.

For each of the task groups there is a configuration parameter definition member in the SNS/TCPaccess PARM dataset. ACP parameters are defined in the *ACPCFGxx*member. API parameters are defined in the *APICFGyy*member. The values for *xx* and *yy* can be determined from the startup command definition member in the SNS/TCPaccess CMND dataset.

ACP Configuration

The ACP configuration statements that are important to TSM client/server performance are the **NETWORK** and the **TIB** statements. Other statements are required to build a complete system.

NETWORK statement

The NETWORK statement describes the interface between the network and the MVS host running SNS/TCPaccess. The complete syntax of the NETWORK statement is shown below:

NETWORK statement syntax

`NETWORK HOST(internet_address) LNID(name1 [name2 ...])`
`[SUBNET(mask) ] [NAME(name) ] [MTU(number) ] [MSSOPT(NEVER | SUBNET | NET`
`| ALWAYS) ] [MSSDEF(number) ] [FWD | NOFWD ] [ARPTABLE(name) ] [LOOP`
`|NOLOOP ]`

The parameters that are important to TSM client/server performance include the following:

MTU(number)

Specifies the local network maximum transmission unit (MTU) size excluding the network header.

MSSDEF(number)

Specifies the default maximum transmission unit (MTU) size sent when no MSS option has been received from the other end of a TCP connection. It must be less than or equal to the MTU for the local network. The standards pecification is 512 bytes and is useful for traffic across an unknown set of networks and gateways. If the networks defined by this NETWORK statement can support a larger size, then set MSSDEF to that size.

MSSOPT(NEVER | SUBNET | NET | ALWAYS)

Specifies when the TCP maximum segment size (MSS) option will be sent. The standards for TCP/IP require that a packet no larger than 512 bytes be sent unless the sender is certain that the receiver can handle it. The MSS option informs the other end of the connection what the limit is. It is recommended to specify MSSOPT(ALWAYS).

Maximum transmission unit size considerations:

- Information is transferred over a TCP connection in direction in discrete packets. Each packet includes a TCP header and an IP header. The header size is independent of the amount of user information included, so the larger the packets sent, the less relative bandwidth is consumed by protocol headers. Also, the TCP software consumes a fixed amount of CPU time for each packet, independent of the packet size, so the larger the packets sent, the less CPU is consumed.
- However, large packets can be fragmented by intervening gateways. Fragmentation and re-assembly of packets are expensive in their use of bandwidth and in CPU time. Therefore, packets sent through gateways to other networks should use the default size unless all intervening gateways and network are known to accept larger packets.

Recommendation
MTU = 4000 if no fragmentation occurs enroute (TR). MTU = 1500 if no fragmentation occurs enroute (EN).

Transport Services Information Block (TIB) statement

The Transport Services Information Block (TIB) statement specifies TCP or UDP transport services parameter values. In most cases, default TIB statements should be included in the configuration for both TCP and UDP.

The complete syntax of the TIB statement is shown below:

TIB statement syntax
`TIB PROTOCOL(TCP | UDP) [MAXQLSTN(listen) ] [MAXQSEND(send)]`

```
 [MAXQRECV(receive) ] [MAXLSEND(send) ] [MAXLRECV(receive) ]
[DEFQSEND(send)]
 [DEFQRECV(receive) ] [DEFLSEND(send) ] [DEFLRECV(receive) ]
[MAXLTSND(send)]
 [MAXLTRCV(receive) ] [TADDRUSE (n1 [:m1] [n2 [:m2]] ... ) ]
[TADDRASSIGN(n1[:m1]
 [n2 [:m2]] ... ) ]
```

The parameters that are important to TSM client/server performance include the following, which should be specified on the TCP TIB statement:

MAXQSEND(send)
: Specifies the maximum number of outstanding API TSEND or TSENDTO requests for each endpoint. During a restore or retrieve operation, TSM may use up to this maximum, or 8 requests, whichever is lower. Limiting this value to a smaller number may reduce client session virtual storage requirements in the TSM and SNS/TCPaccess address spaces, but may also degrade restore or retrieve throughput. This parameter will have no effect on TSM backup or archive throughput, or the client session virtual storage requirements for these operations.

MAXLSEND(send)
: Specifies the maximum number of bytes for all outstanding API TSEND or TSENDTO requests for each endpoint.

MAXLTSND(send)
: Specifies the maximum number of bytes that can be sent in one TSEND or TSENDTO request.

MAXLTRCV(receive)
: Specifies the maximum number of bytes that can be received in one TRECV or TRECVFR request.

TADDRUSE(n1 [:m1] [n2 [:m2]] ...)
: Specifies the range(s) of port numbers that are valid for an API application to use. The port number for the TSM server must be included in this statement.

Recommended TIB example
TIB PROTOCOL(TCP) MAXQSEND(8) MAXLSEND(256000) MAXLTSND(32000) MAXLTRCV(32000) TADDRUSE(1:4095) TADDRASGN(4096:8191)

API Configuration

The API configuration **POOLDEF** statements define pools of control blocks needed to run the Application Program Interface and put limits on API usage.

POOLDEF statement

The complete syntax of the POOLDEF statement is shown below:

POOLDEF statement syntax

POOLDEF NAME(polonium) INITIAL(poolside) MINIMUM(poolside) EXPAND(amount)

NAME(polonium)
Specifies the name of the pool to be defined. One POOLDEF statement must be entered for each of the following pools:

TPCB
one element per transport provider

ARCB
one element per address space using the API

TSQB
one element per task (TCB) using the API

TUCB
one element per **AOPEN** macro issued

EPCB
one element per endpoint opened

SRE
one element per service request made to the API

CBUF
one element per circular buffer built, which is two per endpoint (one for sending and one for receiving)

INITIAL(poolside)
Specifies the initial number of pool elements to be obtained for the pool.

MINIMUM(poolside)
Specifies the minimum number of pool elements to be left in the pool if contraction is performed.

EXPAND(amount)
Specifies the number of elements to be obtained when and if the pool must be expanded.

The default values were used in our lab environment, however, these values may need to be altered depending on system capacity requirements. It is recommended that pool utilization be monitored with the **POOL** command and adjustments made if they appear indicated.

NetWare Clients

NetWare Cache Tuning

The following parameters in NetWare 4.10 (via SERVMAN utility) are tunable and have achieved very good results with TSM.

Table 4. NetWare Cache Tuning Parameters

NetWare Parameter	Default	Range	Recommendation
MAXIMUM NUMBER OF INTERNAL DIRECTORY	100	40-1000	1000

HANDLES			
DIRECTORY CACHE ALLOCATION WAIT TIME	2.2	0.5 - 120	0.5
MAXIMUM PHYSICAL receive PACKET SIZE (TR)	tba	tba	4202
MAXIMUM PHYSICAL receive PACKET SIZE (EN)	tba	tba	1442
MINIMUM DIRECTORY CACHE BUFFERS	tba	tba	2000
MAXIMUM DIRECTORY CACHE BUFFERS	100	20-20000	4000
MAXIMUM CONCURRENT DISK CACHE WRITES	tba	tba	1000
MAXIMUM CONCURRENT DIRECTORY CACHE WRITES	tba	tba	25
MINIMUM PACKET receive BUFFERS	tba	tba	100

MTU

The MTU must be tuned to the appropriate size. In multi-network environments, data traveling from a network with a larger MTU to a smaller MTU, the IP layer must fragment the packet into smaller packets. This will cost the receiving system CPU time to reassemble the fragment packets.

The MTU size is configured by editing STARTUP.NCF (NetWare V3.x) or AUTOEXEC.NCF file (NetWare V4.x) as shown below:

TcpMSSinternetlimit

When data travels to a remote network or a different subnet, the TCPIP. NLMsets the MTU size to the default Maximum Segment Size (MSS) value of 536 bytes. The TcpMSSinternetlimit parameter can be used to override the default MSS value and to set a larger MTU. For NetWare v4.x with TCP/IP v3.0, setting TcpMSSinternetlimit off in SYS:\ETC\TCPIP.CFG will cause the TCPIP. NLM to use the MTU value specified in STARTUP.NCF file (maximum physical receive packet size). Also, note that the TcpMSSinternetlimit parameter is case sensitive. If this parameter is not specified correctly, it will be dropped automatically from the tcpip.cfg file by NetWare.

TCPIP.CFG
TcpMSSinternetlimit off

For NetWare v3.x, Novell patch TCP31A.EXE (for TCP/IP v2.75) can provide the same option.

Solaris Servers and Clients

Performance Recommendations

- The client option ResourceUtilization can be set to value other than default if the backup in the client involves with multiple file spaces(directory, file system) and they are spanning or residing on multiple physical disks. A value of 5 is recommended for the ResourceUtilization option in dsm.sys file. However, for optimal utilization of the TSM environment, you need to take the load of TSM server, network bandwidth , CPU and I/O configuration of the client machine into consideration before using ResourceUtilization option.
- TcpWindowsize 32K, which is set in client's dsm.sys file, is recommended for Solaris client in the FDDI and fast(100mbit) ethernet network environment.
- TcpWindowsize 63K or higher is recommended for Gigabit ethernet network environment. One good way to find the optimal TcpWindowsize value in your specific network environment is to run TTCP program multiple times, with different TcpWindowsize set for each run. The raw network throughput number reported by TTCP can be used as a guide for selecting the best Tcpwindowsize for your TSM server and client. TTCP is freeware which can be downloaded from many SUN freeware web site. Since the default value for TCP xmit and recv buffers are only 8k for Solaris. The default value for tcp_xmit_hiwat and tcp_recv_hiwat need also be changed to the value of TcpWindowsize to avoid any TCP buffer overrun problem. Solaris command "ndd -set" can be used to change the value of these two TCP buffers.

HP-UX Servers and Clients

Performance Recommendations

The following recommendations may help you provide optimized Tivoli Storage Manager performance for your environment.

- Use raw partition for Disk Storage Pool on HP-UX TSM server. Raw partition is recommended because our measurements in the lab show that raw partition volumes offer better backup/restore throughput than when VXFS volumes are used on HP-UX. However, for the data integrity and recoverability of TSM database and recovery log volume, the TSM database and recovery log volume should be allocated in the file systems.
- Use the Tivoli Storage Manager server option txngroupmax 256 and the Tivoli Storage Manager client option txnbytelimit 25600 to maximize the transaction size. A larger transaction size increases the size of the Tivoli Storage Manager server file aggregates. File aggregation provides throughput improvements for many server data movement and inventory functions, such as storage pool migration, storage pool backup, and inventory expiration. A larger transaction

size when using backup directly to tape reduces the number of tape buffer flushes and therefore can improve throughput.

- Use the Tivoli Storage Manager server option uselargebuffers yes to improve throughput and server efficiency for large files. This is the default for Tivoli Storage Manager 3.7.
- Do not set largecommbuffers client option to yes automatically. For both the SUN Solaris TSM client and HP-UX TSM client, setting largecommbuffers to yes may not give you better throughputs all the time. Therefore, experiment with this option with the yes or no in the dsm.sys file to find the appropriate value for your environment.
- Allocate enough separate physical disks to the server to minimize I/O contention.
- For optimal performance of Tivoli Storage Manager functions that drive a significant amount of Tivoli Storage Manager server database activity, including client backup and restore of workloads with many small files, and inventory functions such as inventory expiration and delete filespace, the following actions should be taken: 1) Place the Tivoli Storage Manager server database volumes on the fastest disks. If the write cache exists for the disk adapter that the Database volume is attached to, enable it for the best Database performance. 2) Tune the size of the Tivoli Storage Manager server database buffer pool appropriately. The Tivoli Storage Manager database buffer pool acts as a cache for database pages, so when a page is needed, and the page already exists in the buffer pool, an I/O can be avoided.

Glossary

Acknowledgment - A message sent from one machine to another confirming receipt of data.

Client (TCP/IP) - A client is a computer or process that requests services on the network. A **server** is a computer or process that that responds to a request for service from a client.

Conversation (APPC) - The actual exchange of information between two TP's. Conversations are TP-to-TP links that are carried out ?receivea? session.

CPU Utilization - CPU utilization is computed as total system CPU time divided by the elapsed time.

Data Link Control (DLC)(APPC) - The communications link protocol used to transmit data between two physically linked machines. User data is transmitted inside DLC frames. Token-ring, Ethernet, and SDLC protocols are examples of commonly used DLC' s on SNA networks today.

Elapsed time - All elapsed time is calculated by extracting the time from the TSM activity log for the function being performed. Therefore, the connect time for client backup and restore is not included.

Gigabyte - GByte (or GB) is 1,073,741,824 bytes (two to the thirtieth power) when used in this publication.

Internet address (TCP/IP) - Internet Address is a unique 32 bit address identifying each node in an internet. An internet address consists of a network number and a local address. Internet addresses are represented as a dotted-decimal notation and are used to route packets through the network.

ITR- ITR is the Internal Throughput Rate measured in units of work, for example, files processed, per unit of CPU time.

Kilobyte- KByte (or KB) is 1024 (two to the tenth power) when used in this publication.

KB per CPU second: - The number compares how effectively a single CPU transfers kbytes of data per one CPU busy second. The number is calculated as follows:

```
KB per CPU Second = (Throughput (kb/sec) x 100) /
( number of CPUs x % server CPU utilization)
```

A larger number means CPU is more efficient in transferring the data. This number can be used to compare effectiveness of CPU across different workloads or to compare different CPU types for a given performance evaluation. For SMP systems it is important to understand that this metric applies to the efficiency of a single cpu. The total effectiveness for multiple CPUs in an SMP system working together can be estimated by multiplying by "KB per CPU Second" by the number of available CPUs.

Logical Unit (LU)(APPC) - The "socket" used by a TP to obtain access to an SNA network. LU is the SNA software which accepts and executes the verbs from your Transaction Programs. An LU manages the network on behalf of TP's and is responsible for data routing. A TP gains access to SNA via an LU.

Maximum Transmission Unit (TCP/IP) -

MAXDATARCV (NetBIOS) - The maximum receive data size, in bytes. This is the maximum size of the user data in any frame that this node will receive on a session.

MAXIN (NetBIOS) - The number of NetBIOS message packets received before sending an acknowledgment.

MAXOUT (NetBIOS) - The number of NetBIOS message packets to send before expecting an acknowledgment.

Megabyte - MByte (or MB) is 1,048,576 bytes (two to the twentieth power) when used in this publication.

NETBIOSTIMEOUT (NetBIOS) - The number of seconds that must elapse before a timeout occurs for a NetBIOS send or receive. It is found in the DSM.OPT.

MTU (TCP/IP) - Maximum Transmission Unit. Network parameter which defines the TCP upper limit of the segment size.

NETBIOSBUFFERSIZE (NetBIOS) - The size of the NetBIOS communications buffer in kilobytes. It is found in the DSM.OPT.

Pacing (APPC) - A mechanism used to control the flow of data in SNA. Pacing allows large, fast machines to communicate with smaller, less capable, machines. Pacing is unidirectional. That means that a machine can have a different send window than its receive window.

PACKETS (NetBIOS) - The number of I-frame packet descriptors that the NetBIOS protocol can use to build DLC frames from NetBIOS messages.

Packet (TCP/IP) - A packet refers to the unit or block of data of one transaction between a host and its network. A packet usually contains a network header, at least one high-protocol header and data blocks. Packets are the exchange medium used at the Internetwork layer to send receive data.

Physical Unit (PU)(APPC) - A program on an SNA node that manages physical network resources on behalf of all LU's on that node. For example a PU manages the connections of the node to adjacent nodes. The PU manages the physical data links on behalf of LU's, which in turn manage the sessions or logical needs between nodes.

PIGGYBACKACKS (NetBIOS) - Specifies whether NetBIOS will send and receive acknowledgments piggy backed with incoming data.

Port (TCP/IP) -A port is an end point for communication between applications, generally referring to a logical connection. TCP/IP uses protocol port numbers to identify the ultimate destination within a machine.

Request/Response Unit (RU)(APPC) - A unit of SNA packet which carries user and network control data. It is used to carry user application data from one machine to another and can vary greatly in size.

Segment (TCP/IP) - TCP views the data stream as a sequence of octets or bytes that it divides into segments for transmission. Each segment travels across an internet in a single IP Datagram.

Server CPU Time - This is the total CPU time on the TSM server divided by the total workload in logical MB and expressed as CPU seconds per MB. For measurements with a local TSM client, this includes both the client and server CPU time.

Server CPU Utilization - This is the total CPU time on the TSM server divided by the elapsed time and expressed as a percentage. For measurements with a local TSM client,

this includes both the client and server CPU time. On multiple processor machines, this is an average across all processors.

Server Efficiency - This is the total client workload executed divided by the total CPU time on the TSM server and expressed as KB per CPU second. For measurements with a local TSM client, this includes both the client and server CPU time. This value can be used to compare the CPU efficiency of execution of different tests or workloads.

Server ITR -This is the internal throughput rate (ITR) in workloadfiles processed divided by the CPU time on the TSM server and expressed as files per CPU second. For measurements with a local TSM client, this includes both the client and server CPU time.

SNA session (APPC)- A logical connection between two LU's across an SNA network.

Throughput - Throughput, mentioned in this guide, is total bytes in the workload backed up or restored divided by elapsed time. The total bytes does not include communication protocol and TSM overhead. Note that this differs from the throughput rate which appears on the client screens. The latter is the instantaneous data transfer rate and does not include any delays due to client or server processing or any wait time in communications Throughput is reported in Kbytes per second, where Kbytes i s 1024 bytes. Note that network devices and protocols often quote Kbits or Mbits per seconds (such as a 16 Mbit token ring). For example, 16 Mbits is equivalent to 2 MBytes.

Throughput Ratio: - Denoted with headings "Throughput Ratio" or "Thru Ratio" for table columns. This compares the relative throughput between two performance measurements. For a given description, the first row of each group in the table is used as a reference. The throughput in the following rows is compared to the reference using the following calculation:

```
Throughput Ratio = 100 x (Throughput measured / Reference Throughput)
```

The most prevalent example is how throughput scales when multiple clients are added. A larger throughput ratio means better scaling, conversely smaller number translates to poor scaling, and a number close to 100 indicates little change. This metric acts as a quick comparison tool for the comparison of different environments. This includes but is not limited to:

- adding clients
- adding CPUs (for SMP systems)
- different levels of server codes

Transaction Program (TP)(APPC) - The logical network name used by an application program to communicate with another application program receive an SNA network.

Window (APPC) - A quantity, or count, of data frames which machines remember they have sent. Each window

Window size (TCP/IP) - The number of packets that can be unacknowledged at any given time is called the window size. For example in a sliding window protocol with window size 8 the sender is permitted to send 8 packets before it receives an acknowledgment. of data transmitted requires an acknowledgment to be sent in return.

References

Other Resources

These publications are also relevant as further information sources:

- Tivoli Storage Manager Version 4.2:Technical Guide,SG24-6277
- Tivoli Storage Manager Version 5.1: Technical Guide, SG24-6554-00 Redpiece
- Using Tivoli Storage Manager in a SAN Environment, SG24-6132
- A Practical Guide to Tivoli SANergy SG24-6146
- 3584 UltraScalable Tape Library Planning and Operator Guide, GA32-0408
- 3583 Ultrium Scalable Tape Library Setup and Operator Guide, GA32-0411
- 3581 Ultrium Tape Autoloader Setup, Operator and Service Guide, GA32-0412
- 3580 Ultrium Tape Drive Setup, Operator and Service Guide, GA32-0415
- IBM Ultrium Device Drivers Installation and User s Guide, GA32-0430
- IBM Storage Area Network Data Gateway Module Setup, Operator and Service Guide, GA32-0436
- TSM for AIX Managed System for SAN Storage Agent User s Guide GC36-0001
- TSM for Sun Solaris Managed System for SAN Storage Agent User's Guide, GC36-0002

IBM LTO Related Publications

- Using IBM LTO Ultrium with Open Systems (SG24-6502-00 IBM Redbook Redpiece
- Iplementing IBM LTO Tape in Linux and Windows (SG24-6268 IBM Redbook)
- Designing an IBM Storage Area Network, SG24-5758
- Planning and Implementing an IBM SAN, SG24-6116
- IBM SAN Survival Guide, SG24-6143
- Using Tivoli Storage Manager in a SAN Environment, SG24-6132
- The IBM LTO Ultrium Tape Libraries Guide, SG24-5946

Additional TCP/IP Performance Related Publications

- Douglas E. Comer, "Internetworking with TCP/IP, Vol. I: Principles, Protocols,and Architecture". Prentice-Hall, 1991. ISBN 0-13-468505-9.
- "AIX Version 3.2 for RISC System/6000 Performance Monitoring and Tuning Guide". IBM publication SC23-2365-01.

- http://www.rs6000.ibm.com/support/sp/perf/
- "Networking and Systems Management Services and Support". IBM Publication GG66-3243-00.
- "TCP/IP Version 1.2.1 for OS/2 (Refresh): TCPLAPS". IBM publication SC31-7026.
- "TCP/IP Version 2 Release 2.1 for DOS User's Guide". IBM publication SC31-6153-03.
- "TCP/IP Version 2 Release 2.1 for MVS: User's Guide". IBM publication SC31-6088-02.
- "TCP/IP Version 2 Release 2.1 for OS/2 User's Guide". IBM publication SC31-6076-03.

Referenced Web Sites

- TSM Home Page: http://www.ibm.com/software/tivoli/products/storage-mgr /
- IBM Redbooks: www.redbooks.ibm.com
- Tivoli Storage Manager Products Technical Support. http://www.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html