



# Tips for Tivoli Storage Manager Performance Tuning and Troubleshooting

By Dr. Zong Ling, Tivoli Storage Manager Performance Team  
Version 1.0



## **Copyright Notice**

Copyright IBM Corporation 2007. All rights reserved. May only be used pursuant to a Tivoli Systems Software License Agreement, an IBM Software License Agreement, or Addendum for Tivoli Products to IBM Customer or License Agreement. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without prior written permission of IBM Corporation. IBM Corporation grants you limited permission to make hardcopy or other reproductions of any machine-readable documentation for your own use, provided that each such reproduction shall carry the IBM Corporation copyright notice. No other rights under copyright are granted without prior written permission of IBM Corporation. The document is not intended for production and is furnished "as is" without warranty of any kind. All warranties on this document are hereby disclaimed, including the warranties of merchantability and fitness for a particular purpose.

U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corporation.

## **Trademarks**

IBM, the IBM logo, Tivoli, the Tivoli logo, AIX, Cross-Site, NetView, OS/2, Planet Tivoli, RS/6000, Tivoli Certified, Tivoli Enterprise, Tivoli Enterprise Console, Tivoli Ready, and TME are trademarks or registered trademarks of International Business Machines Corporation or Tivoli Systems Inc. in the United States, other countries, or both. Lotus is a registered trademark of Lotus Development Corporation. Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both. UNIX is a registered trademark of The Open Group in the United States and other countries. C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both. PC Direct is a trademark of Ziff Communications Company in the United States, other countries, or both and is used by IBM Corporation under license. ActionMedia, LANDesk, MMX, Pentium, and ProShare are trademarks of Intel Corporation in the United States, other countries, or both. For a complete list of Intel trademarks, see <http://www.intel.com/sites/corporate/trademarx.htm>. SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC. For further information, see <http://www.setco.org/aboutmark.html>. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Other company, product, and service names may be trademarks or service marks of others.

## **Notices**

References in this publication to Tivoli Systems or IBM products, programs, or services do not imply that they will be available in all countries in which Tivoli Systems or IBM operates. Any reference to these products, programs, or services is not intended to imply that only Tivoli Systems or IBM products, programs, or services can be used. Subject to valid intellectual property or other legally protectable right of Tivoli Systems or IBM, any functionally equivalent product, program, or service can be used instead of the referenced product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by Tivoli Systems or IBM, are the responsibility of the user. Tivoli Systems or IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, New York 10504-1785,

U.S.A.



# Acknowledgement

Special thanks go to Oliver Augenstein/Germany/IBM for his suggestions and team contributions to the topic of TSM for ERP Administration Assistant.

# 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 may be used. Any functionally equivalent program may be used instead.



# About the Tivoli Field Guides

## Sponsor

Tivoli Customer Support sponsors the Tivoli Field Guide program.

## Authors

Those who write field guides belong to one of these three groups:

- Tivoli Support and Services Engineers who work directly with customers
- Tivoli Customers and Business Partners who have experience using Tivoli software in a production environment
- Tivoli developers, testers, and architects

The main team members who wrote this field guide are:

Barbara Wald/Germany/IBM

Charles Nichols/Tucson/IBM

Leonard Ling/San Jose/IBM

Robert Elder/Tucson/IBM

Zong Ling/Almaden/IBM

## Audience

The field guides are written for all customers, both new and existing. They are applicable to external audiences including executives, project leads, technical leads, team members, and to internal audiences as well.

## Types of Field Guides

Two types of Tivoli Field Guides describe how Tivoli products work and how they are used in real life situations:

- Field Guides for technical issues are designed to address specific technical scenarios or concepts that are often complex to implement or difficult to understand, for example: endpoint mobility, migration, and heartbeat monitoring.
- Field Guides for business issues are designed to address specific business practices that have a high impact on the success or failure of an ESM project, for example: change management, asset Management, and deployment phases.

## Purposes

The Field Guide program has two major purposes:

- To empower customers & business partners to succeed with Tivoli software by documenting and sharing product information that provides accurate and timely information on Tivoli products and the business issues that impact an enterprise systems management project
- To leverage the internal knowledge within Tivoli Customer Support and Services and the external knowledge of Tivoli customers and Business Partners

## **Availability**

All completed field guides are available free to registered customers and internal IBM employees at the following Web site:

[http://www.ibm.com/software/sysmgmt/products/support/Field\\_Guides.html](http://www.ibm.com/software/sysmgmt/products/support/Field_Guides.html)

Authors can submit proposals and access papers by e-mail:

[mailto:Tivoli\\_eSupport\\_Feedback@us.ibm.com](mailto:Tivoli_eSupport_Feedback@us.ibm.com)



# Table of Contents

Sponsor.....	vii
Authors .....	vii
Audience .....	vii
Types of Field Guides .....	vii
Purposes.....	vii
Availability .....	viii
1. TIPS FOR TSM PERFORMANCE AND CONFIGURATION.....	2
1.1 <i>Optimizing the server database</i> .....	2
<b>TSM database and log volumes</b> .....	2
<b>Write cache</b> .....	3
<b>TSM mirroring</b> .....	3
<b>TSM server database buffer</b> .....	3
<b>TSM server database size</b> .....	4
1.2 <i>Optimizing server device I/O</i> .....	4
1.3 <i>Using collocation</i> .....	5
<b>Collocation by node</b> .....	5
<b>Collocation by filesystem</b> .....	6
<b>Collocation by group</b> .....	6
<b>Collocation group size</b> .....	6
<b>Automatic collocation</b> .....	6
<b>Active data collocation</b> .....	6
1.4 <i>Increasing transaction size</i> .....	7
<b>Client session transaction size</b> .....	7
<b>Server process transaction size</b> .....	7
<b>Recommended options</b> .....	7
<b>Interaction</b> .....	8
<b>Retry</b> .....	8
1.5 <i>Tuning network options</i> .....	8
<b>TcpNoDelay</b> .....	9
<b>TcpWindowSize</b> .....	9
<b>Tcpbufsize</b> .....	9
<b>Diskbufsize</b> .....	9
<b>Recommended options</b> .....	10

<b>Compression</b> .....	<b>10</b>
1.6 <i>Using LAN-Free</i> .....	11
1.7 <i>Using incremental backup</i> .....	11
1.8 <i>Using multiple client sessions</i> .....	12
<b>Multi-session backup/archive client</b> .....	<b>12</b>
<b>Virtual mount points</b> .....	<b>12</b>
<b>Multiple concurrent backup/restore</b> .....	<b>13</b>
1.9 <i>Using image backup</i> .....	13
<b>Offline/online image backup</b> .....	<b>13</b>
<b>Multiple concurrent backup/restore</b> .....	<b>14</b>
1.10 <i>Optimizing schedules</i> .....	14
<b>2. TIPS FOR TSM PERFORMANCE TROUBLESHOOTING</b> .....	<b>15</b>
2.1 <i>Using TSM server instrumentation</i> .....	15
<b>TSM threads</b> .....	<b>15</b>
<b>Usage strategy</b> .....	<b>16</b>
<b>Platform differences</b> .....	<b>16</b>
<b>TSM server instrumentation categories</b> .....	<b>16</b>
<b>TSM server instrumentation example</b> .....	<b>17</b>
2.2 <i>Using TSM client instrumentation</i> .....	18
<b>TSM client instrumentation categories</b> .....	<b>18</b>
<b>TSM client instrumentation example</b> .....	<b>19</b>
2.3 <i>Using TSM API instrumentation</i> .....	20
2.4 <i>Using the administration assistant of TSM for ERP</i> .....	20
<b>Data collection</b> .....	<b>21</b>
<b>Bottlenecks and balance</b> .....	<b>21</b>
<b>Performance optimization</b> .....	<b>21</b>
2.5 <i>Using operating system tools</i> .....	21
<b>FTP</b> .....	<b>22</b>
<b>UNIX dd</b> .....	<b>22</b>
<b>Others</b> .....	<b>22</b>
<b>3. CASE STUDY</b> .....	<b>23</b>
3.1 <i>Inefficient disk systems</i> .....	23
<b>Problem description</b> .....	<b>23</b>
<b>Problem-solving plan and action</b> .....	<b>23</b>

<b>Problem resolution</b> .....	<b>24</b>
3.2 <i>Bottleneck at tape system</i> .....	25
<b>Problem description</b> .....	<b>25</b>
<b>Problem-solving plan and action</b> .....	<b>25</b>
<b>Problem resolution</b> .....	<b>26</b>
3.3 <i>Problematic configuration of TSM server</i> .....	26
<b>Problem description</b> .....	<b>26</b>
<b>Problem-solving plan and action</b> .....	<b>26</b>
<b>Problem resolution</b> .....	<b>27</b>
3.4 <i>File systems with sensitive mount option</i> .....	27
<b>Problem description</b> .....	<b>28</b>
<b>Problem-solving plan and action</b> .....	<b>28</b>
<b>Problem resolution</b> .....	<b>29</b>
3.5 <i>Slow network</i> .....	29
<b>Problem description</b> .....	<b>29</b>
<b>Problem-solving plan and action</b> .....	<b>29</b>
<b>Problem resolution</b> .....	<b>30</b>
3.6 <i>Unbalanced use of resources</i> .....	31
<b>Problem description</b> .....	<b>31</b>
<b>Problem-solving plan and action</b> .....	<b>31</b>
<b>Problem resolution</b> .....	<b>32</b>
4. SUMMARY .....	33
5. REFERENCES .....	35



# Tips for Tivoli Storage Manager Performance Tuning and Troubleshooting

Tivoli Storage Manager (TSM) performance tuning and troubleshooting are complicated due to the many factors that are involved and yet are highly crucial to leveraging customer satisfaction into a wider acceptance and use of TSM products. This paper is meant to provide insight into the TSM performance configuration and troubleshooting. We introduce topics such as optimizing configuration, tuning parameters, and finding performance bottlenecks, as well as a collection of performance tools including TSM instrumentation and Administration Assistant.

In order to support the inquiries about TSM performance characteristics, such as TSM sizing, capacity, scalability, and others, performance recommendations for TSM are listed in respect to managing the server/client platforms, operating systems, data types, networks, and storage devices.

The target audiences of this paper are advanced TSM implementers. In this paper, in-depth TSM performance tuning knowledge, together with the selective parameters that are tunable for TSM performance, is presented, and TSM Instrument tools are introduced and several customer cases are discussed. This paper is different from another document: TSM Performance Tuning Guide, which is designed mainly for new TSM users and is covered with general TSM performance information associated with a complete set of TSM options.

Because this document is for experienced TSM users and is intended to minimize the overlap with existing documentation, it should be used in conjunction with other TSM publications: *Administrator's Guide*, *Performance Tuning Guide*, *Problem Determination Guide*, see References [1]; and *Administration Center Performance Evaluation Report*, see References [2].

The content in this paper is based on the presentations provided by the TSM performance team in SHARE, Storage Symposium, and IBM Software University. Some content may be available online, such as on the Tivoli Distance Learning website (See References [3]) and IBM Software University 2006 (See References [4]).

This paper is arranged in six sections:

- Section 1 gives a brief overview of the purpose, content, and structure of this paper.
- Section 2 provides useful tips for TSM performance configuration.
- Section 3 shows you the common tools for troubleshooting and exhibits TSM performance problems.
- Customer case studies are described in Section 4 concerning the problem -solving strategies and approaches.
- Section 5 offers a summary of this paper.
- Section 6 contains references.

---

# 1. Tips for TSM performance and configuration

We will use a checklist as a guide for optimizing the performance of the TSM environment and to help configure the server, client, network, and storage device. We could not document all possible situations in this paper, and you may be working on an environment where the optimal solution may differ from what is stated here. In any case, the advice we offer represents a basic starting point from which additional configuration and tuning options can be further explored.

The following is a performance checklist for optimizing your TSM configuration:

1. Optimizing the server database
2. Optimizing server de server vice I/O
3. Using collocation
4. Increasing transaction size
5. Maximizing your tuning network options
6. Using LAN-Free
7. Using incremental backup
8. Using multiple client sessions
9. Using image backup
10. Optimizing schedules

## 1.1 Optimizing the server database

TSM server database performance is critical to the performance of many TSM operations. The following sections highlight several issues that you should consider:

### **TSM database and log volumes**

Using multiple TSM server database volumes provides greater database I/O capacity which can improve overall server performance. Especially during backup, restore, and inventory expiration, the TSM server process may be able to spread its I/O activities over several volumes in parallel, which can reduce I/O contention and increase performance. The recommended number of TSM server database volumes is between 4 and 16, though some environments may require a smaller or larger number than this.

**Note:** In general, having a higher quantity of smaller TSM server database volumes can provide better performance in comparison with having a smaller number of TSM server database volumes with larger size and the same rotation speed. However, in one customer case it was reported that the TSM performance of DBBACKUP or expiration of a TSM server database with 200 x 1 GB volumes may be worse than that of a server with 20 x 10 GB volumes.

TSM performance improves when you use a separate disk (LUN) for each database volume. If more than one TSM server database volume exists on a single physical disk or array, there may be no additional performance gain due to I/O contention. Check the platform performance monitor to determine if the TSM server database disk busy rate is higher than 80%, which would indicate a bottleneck. In general, it is not recommended to place a TSM server database volume on a disk (LUN) that has high sequential I/O activity such as might occur with a TSM disk storage pool volume.

To allocate the TSM server database, you might want to use fast disks with high RPM, low seek time, and low latency, and use disks connected through a SCSI/Fibre channel (not SATA).

For installations that use large disk storage subsystems such as IBM TotalStorage ESS or DS family, consider the location of the TSM server database and recovery log volumes with respect to the RAID ranks within the storage subsystem. You must review the performance information available on a RAID rank basis to determine the optimal placement. Large TSM server environments may want to dedicate a RAID rank to a single TSM server database or recovery log volume to boost performance.

For the reasons of hardware availability, most TSM server database and recovery log volumes are placed on high-availability arrays that use some form of hardware RAID 1, 5, or 10, or use TSM mirroring.

You might not want to use operating system mirroring or software RAID 5 because these methods impact the performance.

Only a single TSM server recovery log volume is necessary to optimize performance, because recovery log I/O tends to be sequential.

### **Write cache**

You might want to use write cache of adapter or subsystem for the disks of TSM server database, TSM recovery log, and TSM storage pools, because of the superior performance of writing random database I/Os to cache. However, this should only be feasible if the disk adapter or subsystem can guarantee recovery, for example; it must be battery backed in the event of a hardware failure (such as loss of power).

Using write cache for all RAID 5 arrays is suggested because there can be a large RAID 5 write penalty if the parity block is not available in cache for the blocks that are being written.

### **TSM mirroring**

The TSM server provides database and recovery volume mirroring which increases the availability of the TSM server, in case of a disk failure. Further availability can be found by using the TSM server database page shadow, which is enabled by the `dbpageshadow yes` server option.

The database page shadow-protects against partial page writes to the TSM server database that can occur if the TSM server crashes, due to a hardware or software failure. This is possible because disk subsystems use 512 Byte sectors, but TSM servers use 4 KB pages. The page shadow allows the TSM server to recover from these situations, which might otherwise require restoring the TSM server database from a recent backup.

You might want to always use the TSM server database page shadow. There is a slight performance impact if you use this feature. You can place the page shadow file in the server install directory.

If the TSM server is using TSM mirroring for the database volumes and using the database page shadow, then it is safe to enable parallel I/Os to each of TSM server database volume copies. Doing so will improve overall TSM server database performance by reducing I/O response time.

### **TSM server database buffer**

In general, the performance of the TSM server should be better if you increase the size of database buffer pool.

Note that the *bufpoolsize* server option is specified in KB. Beginning with TSM 5.3, this option default is set to 32768 (32 MB). The buffer pool size can initially be set to 25% of server real memory or the process virtual memory limit (whichever is lower). For example, if a 32-bit Server has 2 GB RAM, the TSM server database buffer pool size may be set as *bufpoolsize 524288*. The performance benefit may diminish when the buffer size goes beyond 1 GB.

DO NOT increase the buffer pool size if system paging is significant. On AIX, paging may be reduced with the appropriate tuning.

You can issue the *query db f=d* server command to display the database cache hit ratio. As a general rule, the TSM server database cache hit ratio should be greater than 98%.

### **TSM server database size**

The way that you use the size or space of the TSM server database impacts TSM scalability. The consideration for multiple TSM server instances is often motivated by the size of the TSM server database.

To estimate the space usage of a TSM server database, you can consider that the TSM server database uses the space as 3-5% of the size of the total file system data. Also you can consider that in the TSM server database space, we generally use the 600 bytes per object figure for the primary copy of an object. For HSM objects with backups, those are really two different objects to the TSM server, so multiply the number of bytes required per object by 2. If you keep a copy of an object (in a TSM copy pool), that is an additional 200 bytes for each copy of each object. Additional versions of backup objects also count as new objects - 600 bytes per object. These numbers could go up or down depending on the directory depth and length of the path and file names of the data that is stored.

A common practice is to have the average size of the TSM server database be around 40 -60 GB, although it could be bigger (such as 200+ GB as reported in some customer cases). For the larger sized database, more time may be cost to periodically maintaining the database optimization. For instance, it may take about one hour per each GB that the TSM server database has for the TSM server *auditDB* process to run. And during the period of maintenance, the TSM server does not perform any normal operations. Note that database audit is not a 'normal' maintenance requirement, but an exception that needs to be done only in certain specific cases.

We suggest that as a TSM server database starts approaching 80 -100 GB, that it is better to configure multiple TSM servers on one capable machine. For procedures on configuring multiple TSM servers, see References [5].

## **1.2 Optimizing server device I/O**

The TSM server performance depends upon the system I/O throughput capacity. Any contention of device usage when data flow move s through device I/O impacts TSM throughput performance. There are several performance strategies that might help in manag ing the devices.

You might want to study the system documentation to learn which slots use which particular PCI bus and put the fastest adapters on the fastest busses. For best LAN backup -to-disk performance, you might want to put network adapters on a di fferent bus than the disk adapters.



For best disk-to-tape storage pool migration performance, put disk adapters on a different bus than on the tape adapters.

Since parallelism allows multiple concurrent operations, it is reasonable to use *multiple* Busses, Adapters, LANs and SANs, Disk subsystems, Disks, Tape drives, and TSM client sessions.

When you use a DISK storage pool (random access), it is better to define multiple disk volumes and one volume per disk (LUN). If using a FILE storage pool (sequential access), it is helpful to use multiple directories in the device class and one directory per disk (LUN).

Consider configuring enough disk storage for at least one day's backups and configuring disk subsystem LUNs with regard for performance.

**Note:** It is important to separate the disk I/O for TSM storage pools from the disk I/O for the TSM server database, because the TSM server database I/O is significantly degraded when heavy TSM storage pool I/O is also initiated on the same set of disks.

## 1.3 Using collocation

TSM restore performance of large file servers can be poor if the backup data is not collocated. Usually this is due to the backup data being spread over a huge amount of tape volumes and thus requiring a large number of tape mounts. TSM collocation is designed to maintain optimal data organization for faster restore, backupset generation, export, and others because of less mount and locate time for tape and lower volume contention during multiple client restores. Note that more storage volumes may be required when using collocation.

The following storage pool parameters (for sequential -access only) specify the types of TSM collocation:

No – data is written to any available volume

Node – data is written to any volume with data for this node

Filespace – data is written to any volume with data for this filesystem

Group – data is written to any volume with data for this group of nodes

### Collocation by node

To avoid volume contention, the nodes that have a low chance of restore at the same time can be collocated together by using *collocation=node*.

If storage pool migration has to run during the backup window, since all data that are collocated can be moved at one time, the nodes that back up to disk at the same time should be collocated together in order to reduce volume mounts.

Faster tape volume reclamation can occur when database data is separated from file server data, either by using collocation by node or by keeping data in separate storage pools. This is because database backups are typically large enough to fill the tape volumes and whole tape volumes can expire at one time. This leads to tape volumes being reclaimed with no data movement. If file server data is mixed in with database data in a single non-collocated storage pool, volumes can contain small amounts of file server data that may not expire at the same time as the database data, thus leading to the volumes having to be mounted and read during reclamation processing.

Collocation by node is best at restore time, but not optimized for mult i-session restore. Also, collocation by node requires the highest volume usage and the highest number of mounts for migration and reclamation.

### **Collocation by filesystem**

Restoring a large file server with multiple large file systems requires multiple restore commands. If the backup data for these multiple file systems is collocated by node, and thus contained on the same sequential media volumes, there is contention between the restore processes for access to these volumes, which also impacts performance.

The file servers with two or more large file systems should use a TSM storage pool that is collocated by filesystem by using *collocation=filesystem*. Collocation by filesystem honors the “virtual” file systems that are defined using TSM client option *virtualmountpoint*.

### **Collocation by group**

You might want to use collocation by group (*collocation=group*) for copy storage pools where volumes are taken offsite, if the groups defined are sufficiently large enough so that only a small number of volumes must be updated on a daily basis.

You might want to use collocation by group for primary storage pools on tape. Nodes that are not grouped are collocated by node.

### **Collocation group size**

The optimal collocation group size depends on the (tape) volume size and the number of mount points (tape drives) available for restore of a single node.

The volume size depends on the tape drive technology, tape media technology, and the data characteristics (such as compressibility). The PERL script of *fullvolcapacity* can be used to determine the average full volume capacity for each sequential -access storage pool on the server. The Administration Center wizard determines the volume size automatically by using the first sequential-access storage pool found in the default management class destination.

The volume size should be multiplied by the number of available mount points or tape drives available for restore of these nodes. It is necessary to consider the total number of tape drives that the TSM server has available and the number that might be in use for other processes (database backup, storage pool migration, node backups, other restores) concurrent with a restore. The Administration Center wizard uses a value of 4.

### **Automatic collocation**

If you have no knowledge about the relationships between nodes, no knowledge about the node data growth rates, or no knowledge about how existing data is collocated, you might want to use automatic methods to collocate the groups. Use automatic methods first, then manually fine -tune the types of collocation.

### **Active data collocation**

Beginning with TSM 5.4, a new storage pool called Active -Data Pool (ADP) is available to collocate the active data and thus improve TSM restore performance significantly. By using this feature during the restoring process, data can be directly restored from an ADP which only stores active data. Therefore, the overhead of distinguishing active/inactive data in a primary storage pool can be avoided.

The following are two methods that you can use to collocate active data into an ADP:

- For the legacy data in a primary storage pool, issue the *COPY ACTIVE DATA* TSM server command so that the active data in that primary storage pool can be extracted out and migrated to the ADP.
- For the new data coming from TSM backup sessions, configure the ADP and the primary storage pool appropriately so that the active data can be simultaneously backed up to the ADP and the primary storage pool.

## 1.4 Increasing transaction size

Generally, by increasing TSM transaction size you can increase the throughput of client backup and server data movement. The following sections offer some helpful recommendations:

### Client session transaction size

The following options are designed to control TSM client/server session transaction size for the backup/archive client only:

*txngroupmax* specifies the maximum number of objects (files and/or directories) included in a client session transaction. For TSM 5.2 and later, this option is set globally for TSM server and can be set individually for each node by issuing the *update node* command.

*txnbytelimit* specifies the maximum number of object size in KB that are included in a client session transaction. A single file exceeding this size is always processed as a single transaction.

TSM for Space Management (HSM), TSM for Databases, TSM for Mail, TSM for ERP, and other TSM applications will typically send a single object (file or database) in a transaction.

### Server process transaction size

The following options are designed to control TSM server data movement transaction size which can influence the performance of storage pool migration, storage pool backup, storage pool restore, reclamation, and move data functions:

*movebatchsize* specifies the maximum number of objects (server physical bitfiles) included in a server data movement transaction.

*movesizethresh* specifies the maximum number of object size in MB that are included in a server data movement transaction.

### Recommended options

The following server options are recommended to setup (TSM 5.3 defaults):

- *txngroupmax 256*
- *movebatchsize 1000*
- *movesizethresh 2048*

Also the following backup/archive client option:

- *txnbytelimit 25600*

For the nodes that back up small files directly to tape, it is suggested that the node definition be updated with the `UPDATE NODE nodename txngroupmax=4096` command. Also, the client option can be set as `txnbytelimit 2097152`.

Note that increasing transaction size may be mostly helpful to improve the performance when moving data to tape, including backup storage pool from disk to tape. It may not be beneficial when moving data to disk, which may cost more TSM server recovery log space.

You might want to increase the TSM server recovery log size. Currently, 4 GB is sufficient for many installations with 13.5 GB being the maximum.

## Interaction

To optimize performance, minimal interaction and information display should be configured when backing up or restoring a large number of files. The following options are designed to control the amount of interaction required and information displayed to the user during a client backup or restore session:

*tapeprompt* specifies whether mounting a tape should be waiting, if it is required for a backup, archive, restore, or retrieve process, or if it is to be prompted for a choice. Specifying *no* indicates that no prompt is made and the process waits for the appropriate tape to be mounted. Tape prompting does not occur during a scheduled operation regardless of the setting of the *tapeprompt* option.

*quiet* limits the number of messages that display during processing. This option also affects the amount of information reported in the TSM client schedule log and the Windows client event log. If *quiet* is not specified, the default option, *verbose*, is used. By default, information about each file that is backed up or restored is displayed. Errors and session statistics are displayed regardless of whether *quiet* or *verbose* is specified.

## Retry

A retry occurs when processing is interrupted. Because of retry, increasing TSM transaction size may degrade throughput performance when the data must be frequently sent again. The retry information is found by checking client session messages or schedule log file (*verbose*).

In order to avoid retry, use client option *compressalways yes* and *tapeprompt no* or *quiet*.

To reduce the number of retries, you can set up the client option *changingretries*.

It is reasonable to scheduling backup/archive processes when the files are not in use or using exclude options to exclude files likely to be open. On Windows client, using Open File Support is suggested. To change how open files are managed, use the copy group *serialization* parameter.

## 1.5 Tuning network options

TSM performance is heavily impacted by network configuration because TSM moves data through a network. Some general recommendations on tuning the network options for TSM performance are listed here.

When using Fast Ethernet (100 Mb/sec), make sure that the speed and duplex settings are set correctly. Do not rely on auto-detect, as it has been found to be a frequent cause of poor network performance. If the client adapter, network switches or hubs, and server adapter all support 100 Mb duplex, choose this setting for all.

When using Gb Ethernet, if the client adapter, network switches, and server adapter all support jumbo frames (9000 Byte MTU), then enable this feature to provide improved throughput with lower host CPU usage.

The following options are designed to control TSM data movement through a network.

### **TcpNoDelay**

The TSM *tcpnodeelay* parameter specifies whether TCP/IP will buffer successive small (less than the network MSS) outgoing packets. This should always be set to *yes*.

### **TcpWindowSize**

The TCP window size is a parameter provided in the TCP/IP standard that specifies the amount of data that can be buffered at any one time on a session. If one session partner reaches this limit, then it cannot send more data until an acknowledgement is received from the other session partner. Each acknowledgement includes a window size update from the sending partner. A larger TCP window size may allow the sender to continue sending data and thus improve throughput. A larger TCP window size is particularly useful on reliable (no lost packets), long distance, or high latency networks.

The TSM *tcpwindowsize* option is available on both client and server, and is specified in KB. Many platforms require tuning the operating system in order to use a TCP window size larger than 65535 Bytes (64 KB - 1). If this tuning is not done and the TSM *tcpwindowsize* option is set to 64 or larger, there may be a performance impact caused by the operating system choosing to use a smaller value than that requested by TSM. This is why using *tcpwindowsize 63* is recommended. If the client backups are over a network that might benefit from a large TCP window size, use *tcpwindowsize 63* first, and then experiment with larger values.

Note that for large TCP window size to be used between TSM client and TSM server, both must support this feature, which is also referred to as TCP window scaling, or RFC 1323 support. For the support information of TCP window scaling, see Chapter 2 of TSM *Performance Tuning Guide* from Reference [1].

### **Tcpbufsize**

The TSM *tcpbufsize* is designed to specify the size of the buffer used for TCP/IP *send* requests. During a restore, client data moves from the TSM session component to a TCP communication driver. The *tcpbufsize* option is used to determine if the server sends the data directly from the session buffer or copies the data to the TCP buffer.

A number of the buffer size in KB can be used to force the server to copy data to its communication buffer and flush the buffer when it fills.

**Note:** This option is not related to the *tcpwindowsize* option.

### **Diskbufsize**

The TSM *diskbufsize* option is designed to specify the maximum disk I/O buffer size in KB that the TSM client may use when reading files. For TSM backup/archive or HSM migration client, optimal performance can 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 or even degrade performance.

Note that in TSM 5.3, the previous *largecommbuffers* option is replaced by the *diskbufsize* option.

## Recommended options

The following TSM server options are recommended to set up (TSM 5.3 defaults, new for TSM 5.3 z/OS server):

- *tcpwindowsize* 63
- *tcpnodelay* yes
- *tcpbufsize* 32 (not used for Windows)

As well as the following TSM client options (TSM 5.3 defaults):

- *tcpwindowsize* 63
- *tcpnodelay* yes
- *tcpbufsize* 32
- *diskbufsize* 32, 256 (for AIX, if set *enablelanfree* no), 1023 (for HSM, if set *enablelanfree* no)

## Compression

The TSM client *compression* option helps to instruct the TSM client to compress files before sending them to the TSM server.

Compressing files reduces the file data storage space and can improve throughput over slow networks with a powerful client. The throughput, however, can be degraded when running on a slow client system using a fast network, because software compression uses significant client CPU resources and costs additional elapsed time.

If the option is set as *compressalways* yes, compression continues even if the file size increases. To stop compression if the file size grows and to resend the file uncompressed, set the *compressalways* no option.

If the option is set as *compression* yes, the compression processing can be controlled in the following ways:

- Use the *include.compression* option to include files within a broad group of excluded files for compression processing.
- Use the *exclude.compression* option to exclude specific files or groups of files from compression processing, especially for the objects that are already compressed or encrypted, such as gif, jpg, zip, mp3, and etc.

The preferred setup for the client options are as follows:

- For fast network AND fast server, set *compression* no
- For LAN-Free with tape, set *compression* no
- For slow network OR slow server, set *compression* yes
- Normally, set up *compressalways* yes

Compression statistics can be monitored for each client and the options can be adjusted as needed.

If the TSM client compression and encryption is used for the same file during backup, the file is first compressed and then encrypted, as this will result in a smaller file. On restore, the file is decrypted first and then decompressed.

Note that server storage can be saved by using tape hardware compaction rather than TSM client compression.

## 1.6 Using LAN-Free

TSM “LAN-Free implementation” attempts to offload LAN traffic and get better TSM server scalability due to reduced I/O requirements. Therefore, by using LAN-Free, the higher backup/restore throughput is possible in some circumstances where large objects are moved through the SAN.

Note that TSM LAN-Free needs LAN configuration to enable the meta-data communication through LAN between storage-agent (client) and server. Thus, intuitively, it is not designed for effectively moving small files. For instance, if TSM LAN-Free moves a very small file that is x Bytes through SAN, the file’s meta-data in a size of xxx Bytes must go through LAN. Clearly therefore, it is inefficient to use TSM LAN-Free moving a huge amount of small files. In general, TSM LAN-Free should be used where the SAN bandwidth requirement is significantly greater than LAN. If using TSM LAN-Free to move many small files, then slower performance has to be tolerated.

The usage of LAN-Free is best with “TSM for ...” data protection products and API clients that backup/restore large objects, such as TSM for Mail, TSM for Databases, TSM for ERP.

LAN-Free performance is improved in TSM 5.2 though the following improvements:

- A reduced overhead between storage agent and server on meta-data movement.
- Multi-session scalability is getting better.
- Storage agent tape volume is better managed.

In TSM 5.3, LAN-Free performance improvements are shown on the following:

- CPU usage is reduced, especially for API clients.
- *lanfreecommmethod sharedmem* is used for all TSM 5.3 LAN-Free clients, including AIX, HP-UX, HP-UX on Itanium2, Linux, Solaris, and Windows.

## 1.7 Using incremental backup

Because of TSM’s progressive incremental backup mechanism, it is not necessary to do weekly full backups for file servers. TSM incremental backup compares the client file system with TSM server inventory to determine the files and directories that are new or changed. Then TSM incrementally backs up the new or changed files and directories. TSM incremental backup also expires those deleted files and directories. Therefore, no unnecessary data will be backed up, no need to restore the same file multiple times, less network and server bandwidth are needed, and less server storage are occupied.

For effective incremental backup, use the journal-based service which is available on both Windows and AIX platforms. Journal-based backup uses a journal service that runs continuously on the TSM client machine to detect files or directories that are changed and maintain that information in a journal for use during a later backup. When you use the journal, during an incremental backup the client file system is not scanned and the file and directory attributes are not compared to determine which files are to be backed up. Eliminating these operations means

that journal-based incremental backup can perform much faster than standard full incremental backup and use significantly less virtual memory.

Journal-based backup is multi-threaded for both change notification and backup operations. Started from TSM 5.3, the journal service supports multiple incremental backup sessions concurrently.

Journal-based backup is installed using the client GUI setup wizard and is supported on all current 32-bit Windows clients, as well as in clustered configurations. The journal options are specified in the `tsmjbbd.ini` file in the TSM client install directory. The default configuration parameters are usually optimal. Just add the appropriate file systems to be monitored and the appropriate entries in the journal exclude list.

For further information on configuring the journal service, refer to *IBM Tivoli Storage Manager for Windows Backup-Archive Clients Installation and User's Guide, Appendix C, Journal Service Configuration* (See References [6]).

Started from TSM 5.4, the method with client option *memoryefficientbackup diskcachemethod* is available to improve performance of TSM backup/archive client for executing progressive incremental backup. In comparing to using one server query per directory (*memoryefficientbackup yes*), this method is using disk caching of the inventory data, therefore reducing the amount of virtual memory used by the TSM backup/archive client process and making it possible to back up much larger file systems as defined by the number of files and directories in that file system. Note that using disk caching will not increase the workload on the TSM server.

For more information on incremental backup memory requirements, see References [9].

## 1.8 Using multiple client sessions

TSM backup/archive client can run multiple parallel sessions which can improve backup and restore performance.

### **Multi-session backup/archive client**

Multiple sessions with the server can be enabled by setting the TSM client option *resourceutilization* during backup or restore. Substantial throughput improvements can be achieved in some cases, but is not likely to improve incremental backup of a single large file system with a small percentage of changed data.

If backup is direct to tape, the client node maximum mount points allowed parameter, *maxnummp*, must also be updated at the server by issuing the *update node* command. This specifies the maximum number of tape volumes that can be mounted for this node.

For restore, only one session is used to restore all data in DISK storage pools. For data in sequential storage pools (including FILE device class), as many sessions as there are sequential volumes with data to be restored are used. This is subject to the limits imposed by the node *maxnummp* parameter and the number of mount points or tape drives in the device class.

### **Virtual mount points**

The client option *virtualmountpoint dir\_name* can be used to inform the TSM server that the *dir\_name* directory and all data under it should be considered as a separate "virtual" file system



for backup purposes. This allows better granularity of the backup processing, such as reducing backup process virtual memory usage and enabling multiple parallel backup processes.

You want to create virtual mount points carefully so that each “virtual” file system does not contain more than several million files. This option is only available on UNIX platforms and has no impact on the applications or users of the file system.

## Multiple concurrent backup/restore

Scheduling incremental backups for multiple file systems concurrently on one TSM client system is feasible with any of the following methods:

- Using one TSM node name, running one TSM client scheduler, and using the TSM client option *resourceutilization 5* or greater with multiple file systems included in the schedule or domain specification.
- Using one TSM node name, running one TSM client scheduler, and scheduling a command that runs a script on the client system that includes multiple TSM command line client statements (example: using multiple *dsmc* commands).
- Using multiple TSM node names and running one TSM client scheduler for each node name, in which each scheduler uses a unique client options file.

You might want to configure multiple sessions for “TSM for ...” data protection products, as each has its own configuration options, and to configure multiple sessions for the TSM backup/archive client by setting client option *resourceutilization* to 5 or more or by running multiple client instances in parallel, each on a separate file system.

Multi-session restore should only be used if the restore specification contains an unqualified wildcard (example: *e:\users\\**) and the restore data is stored on multiple sequential storage pool volumes.

## 1.9 Using image backup

TSM image backup can be used to optimize large file system backup and restore performance by using sequential block I/O and avoiding file system overheads, such as *file open()* and *close()*. Therefore, TSM image backup can approach the speed of hardware device .

The TSM *backup image* command can be used to create an image backup of one or more volumes on the client system. TSM API must be installed to use the *backup image* command.

Using TSM image backup can restore the file system to a point-in-time image backup, to a most recent image backup, and to a most recent image backup with changes from the last incremental backup.

### Offline/online image backup

An offline image backup prevents read or write access to the volume by other system applications during the operation. Offline image backup is available for AIX, HP -UX, and Solaris clients.

For Linux86 and Linux IA64 clients, TSM performs an online image backup of file systems residing on a logical volume created by Linux Logical Volume Manager during which the volume is available to other system applications.

For Windows clients, if Logical Volume Snapshot Agent (LVSA) is installed and configured, TSM performs an online image backup during which the volume is available to other system applications. To specify the LVSA snapshot cache location, you can use the *snapshotcachelocation* and *include.image* options.

## Multiple concurrent backup/restore

Scheduling image backups for multiple file systems concurrently on one TSM client system is feasible with either of the following two methods:

- Using multiple TSM node names and running one TSM client scheduler for each node name, in which each scheduler uses a unique client options file.
- Using one TSM node name, running one TSM client scheduler, and scheduling a command that runs a script on the client system that includes multiple TSM command line client statements (example: using multiple *dsmc* commands).

For the best image backup performance, use LAN-Free with tape and use parallel image backup/restore sessions for the clients with multiple file systems.

## 1.10 Optimizing schedules

To reduce resource contention and improve performance of TSM operations, a common strategy is to create schedules with minimal overlap to some TSM operations.

The TSM operations that should be scheduled in different time periods include client backup, storage pool backup, storage pool migration, database backup, inventory expiration, and reclamation.

**Note:** TSM client session starting times can be staggered over the schedule window by using *set randomize percent*. Automatic expiration can be disabled by using server option *expinterval 0*. It is better to define an administrative schedule for expiration at a set time.

Scheduling improvements from TSM 5.3 include calendar-type administrative and client schedules and new commands that include *migrate stgpool* and *reclaim stgpool*.

---

## 2. Tips for TSM performance troubleshooting

Some tools including TSM performance instrumentation, Administration Assistant, and several commands of operating systems are introduced here for diagnosing TSM performance problems. The concepts, usages, and output information of these tools are illustrated.

TSM performance instrumentation is embedded into the TSM code and is available on both the TSM server and the TSM client. There is minimal performance impact during instrumental periods.

### 2.1 Using TSM server instrumentation

With TSM performance instrumentation, most TSM operations that hold up throughput performance are tracked, such as disk read/write, network receive/send, and tape read/write. TSM instrumentation results -- the elapsed times and etc. of TSM operations are recorded and stored in memory until the instrumentation is ended.

TSM server instrumentation is designed to identify the elapsed time of TSM server I/O operations. It can be easily started by using the TSM server command:

```
INSTrumentation Begin [MAXThread= nnnnn]
```

And can be stopped by using the TSM server command:

```
INSTrumentation End > file
```

Where *nnnnn* is the maximal number of threads specified in the RSM server instrumentation.

When TSM server instrumentation is stopped, the output is collected and stored in the *file*.

TSM server instrumentation can also be started/stopped by issuing the following command line on the administrative client console:

```
dsmadm -id=id -pass=pass inst begin
```

```
dsmadm -id=id -pass=pass inst end > file
```

As well as using command redirection with TSM storage agents:

```
dsmadm -id=id -pass=pass agentname: inst begin
```

```
dsmadm -id=id -pass=pass agentname: inst end > file
```

**Note:** The TSM administrator must have system privilege.

#### TSM threads

TSM operations are tracked on a thread-by-thread basis.

Most TSM sessions/processes use more than one thread. TSM backup, for example, may use at least four threads, such as the following:

- SessionThread – that receives data from client
- SsAuxThread – that takes the data and passes to disk or tape
- AgentThread – that writes the data to tape
- DiskServerThread – that writes the data to disk

Because TSM is multi-threaded in operations, the TSM server may have hundreds of threads active at a given time. Thus, the TSM server command *show threads* is often used for collecting a list of threads at any given time.

Note that all threads can operate on different CPUs.

## Usage strategy

You might want to start TSM server instrumentation just before starting the operation monitored. Do this by using a TSM administrative client macro. For TSM 5.3 and later, session and process numbers are included in the output if the thread is started while instrumentation is active.

For the sake of practicality, it is preferable to use TSM server instrumentation for 1 to 30 minutes because a lot of information can be generated in longer periods, and it is harder to diagnose a problem among a larger number of threads.

You must match up the multiple threads for a given session or process. The session or process numbers in the instrumentation data can be used along with the output of the *show threads* command during the operations. Match the threads based on the amount of data moved.

To identify the performance bottlenecks, look at those threads with most of the time in areas other than 'Thread Wait.' That is, most likely, the source of the performance problem.

## Platform differences

TSM server instrumentation output data is slightly different, depending on the platform.

On z/OS systems, thread IDs are not reused like other platforms ; for example, thread IDs increase over time through the server lifetime. Therefore, issue the *show threads* command and note the current high water mark thread ID. It is better to add 1000 to the high water mark and use this as the *maxthread* parameter on the start command. For example: *inst begin maxthread=5000*. Note that *maxthread=xxxx* is only needed when invoking TSM server instrumentation on z/OS platform.

On UNIX, only 1 thread (called DiskServerThread) does I/O activity to any disk storage pool volume, so that a disk volume centric view is provided. From this you might find it harder to get complete disk operation statistics.

On Windows, any thread can perform I/O activity on a disk storage pool volume (such as SsAuxThread for backup). Therefore, a process/session oriented view is provided. Nevertheless, it may be harder to identify disk contention issues. Note that Windows timing statistics only have about 15 millisecond granularity.

## TSM server instrumentation categories

The following list includes the operations that TSM server instrumentation tracks, along with a brief description:

Disk Read                      - reading data from disk

Disk Write	- writing data to disk
Disk Commit	- finishing fsync or other system call to ensure that writes are complete
Tape Read	- reading data from tape
Tape Write	- writing data to tape
Tape Locate	- locating data to a tape block
Tape Commit	- synchronizing tape to ensure data is written from device buffers to media
Tape Data Copy	- copying data to tape buffers (in memory)
Tape Misc	- doing other tape operations (open, rewind, and etc...)
Data Copy	- copying data to various buffers (in memory)
Network Recv	- receiving data on a network
Network Send	- sending data on a network
Shmem Read	- reading data from shared memory buffer
Shmem Write	- writing data to shared memory buffer
Shmem Copy	- copying data to/from shared memory segment
Namedpipe Recv	- receiving data on a named pipe
Namedpipe Send	- sending data on a named pipe
CRC Processing	- computing or comparing CRC values
Tm Lock Wait	- acquiring transaction manager lock
Acquire Latch	- acquiring a database page (from disk or bufferpool)
Thread Wait	- waiting on some other thread
Unknown	- spending on something that is not tracked as above

### **TSM server instrumentation example**

Here is an example of TSM server instrumentation output.

```
Thread 33 AgentThread parent=0(AIX TID 37443)11:09:37.024 -->11:14:27.280
Operation          Count Tottime Avgtime Mintime Maxtime InstTput Total KB
-----
Tape Write         2125   6.191   0.003   0.000   0.010  87556.7   542117
Tape Commit        15    25.505   1.700   0.000   1.764
```

```

Tape Data Copy    2123    1.830    0.001    0.000    0.001
Thread Wait      2175 256.671    0.118    0.000   42.869
-----
Total                290.255                1867.7   542117

Thread 32 SessionThread parent=24(AIX TID 27949)11:10:19.630 -->11:14:13.603
Operation          Count Totttime Avgtime Mintime Maxtime InstTput Total KB
-----
Network Recv     127329 189.952    0.001    0.000    0.415    2865.9   544385
Network Send         36  0.001    0.000    0.000    0.000         0.0         0
Thread Wait       2187  25.552    0.012    0.000    1.766
-----
Total                233.972                2326.7   544386

```

## 2.2 Using TSM client instrumentation

TSM client instrumentation is designed to identify the elapsed time of TSM client operations. You can activate it by using the client command line when launching client sessions:

**-testflag=instrument:detail**

You can also enable it by inserting the following line into the client option file (dsm.opt):

**testflag instrument:detail**

TSM client instrumentation output is appended to a file in the directory specified with the DSM\_LOG environment variable. For TSM 5.2, it is the dsminstr.report file, and for TSM 5.3 and later, it is the dsminstr.report.pPID file.

Note that TSM client instrumentation functions for backup/archive client only (example: it is not applicable for in API or TSM for ... products) and is available for command line client and scheduler only (example: it can not be enabled in GUI or web client). To activate TSM client instrumentation on schedule, the scheduler may need to be restarted after editing the option files.

It is feasible to terminate the TSM client instrumentation by canceling the client session from the TSM server console, to get results without waiting for the client session to finish.

### TSM client instrumentation categories

The following list includes the operations that TSM client instrumentation tracks, along with a brief description:

- Query Server Dirs (TSM 5.4) - receiving server inventory directories for incremental backup
- Query Server Files (TSM 5.4) - receiving server inventory files for incremental backup
- Process Dirs - scanning for files to back up
- Cache Examine - scanning local disk cache db for files to expire (TSM 5.4)

- Solve Tree                   - determining directory structure
- Compute                    - computing throughput and compression ratio
- BeginTxn Verb              - building transactions
- Transaction                - doing file open, close, other misc. operations
- File I/O                    - doing file read and write
- Compression                - compressing and uncompressing data
- Encryption                 - encrypting and decrypting data
- CRC                         - computing and comparing CRC values
- Delta                       - processing adaptive subfile back up
- Data Verb                  - sending and receiving data to/from the server
- Confirm Verb               - responding time during backup for server confirm verb
- EndTxn Verb                - doing server transaction commit and tape synchronization
- Other                       - spending on everything else

### TSM client instrumentation example

Here is an example of TSM client instrumentation output.

Thread: 3 Elapsed time = 10258.410 sec

Section	Actual(sec)	Average(msec)	Frequency used
Query Server Dirs	67.347	67347.2	1
Query Server Files	1030.387	1030386.9	1
Process Dirs	8422.164	59.9	140610
Cache Examine	726.443	726443.2	1
Solve Tree	0.000	0.0	0
Compute	0.000	0.0	0
BeginTxn Verb	0.000	0.0	0
Transaction	0.000	0.0	0
File I/O	0.000	0.0	0
Compression	0.000	0.0	0
Encryption	0.000	0.0	0
CRC	0.000	0.0	0
Delta	0.000	0.0	0
Data Verb	0.000	0.0	0
Confirm Verb	0.000	0.0	0
EndTxn Verb	0.000	0.0	0
Sleep	0.062	61.8	1
Thread Wait	0.066	65.9	1
Other	11.940	0.0	0

## 2.3 Using TSM API instrumentation

TSM API instrumentation is designed to identify the elapsed time of the TSM API client operations. It is new to TSM 5.3 and can be enabled by putting the following line into TSM client option file:

```
testflag instrument:api
```

The output is appended to a file in the directory specified in the DSM\_LOG environment variable:

```
dsminstr.report.pPID
```

Note that TSM API instrumentation cannot be enabled from the command line.

TSM API instrumentation is used in “TSM for ...” products that use the TSM API.

TSM API instrumentation categories are different from, but similar to that of regular TSM client instrumentation.

The following is an example of TSM API instrumentation output:

```
TSM Client final instrumentation statistics: Thu Aug 03 16:35:59 2006
Instrumentation class: API
Completion status: Success
-----
Detailed Instrumentation statistics for
Thread: 4456 Elapsed time 506.344 sec
Section          Actual (sec)    Average(msec)    Frequency used
-----
Waiting on App           76.930           2.5              30404
API Send Data           429.383          14.1             30396
API Query                0.000            0.0              0
API Get Data            0.000            0.0              0
API End Txn            0.031            31.0             1
API Misc                0.000            0.0              1
Other                   0.000            0.0              0
-----
```

## 2.4 Using the administration assistant of TSM for ERP

A performance monitor is available for the enterprise environment using TSM for Enterprise Resource Planning (ERP), on the platforms of UNIX, Linux, and Windows. The Administration Assistant is available to help observe, tune, and optimize the performance of TSM, as well as TSM for ERP. The Administration Assistant installation package is available as a file in the CD image of TSM for ERP or you can download it from the IBM FTP site (See References [6]).



## Data collection

The Administration Assistant uses the functionality called “performance sensors” to observe the incoming and outgoing data streams of a TSM environment. It can then detect the bottleneck location from the performance data on client disk I/O and network throughput. The Administration Assistant function “View Performance Data” provides a graphical representation of the data throughput at any point in time during backup or restore. Aligned with this, the bandwidth utilization rates and the idle time of the client disk and the network threads can be collected and displayed.

## Bottlenecks and balance

If you have a client disk bottleneck, you can observe on the graphic presentation that the data processed by the network and the TSM server is faster than what is read from the client disk. As a consequence, overall throughput is limited by the client disk I/O rate. The network thread may be idle and the capacity of both network and storage device may not be effectively used. If tape devices are used as the storage, then the tapes are not kept in streaming mode any more in this situation.

If you have a bottleneck at network or TSM server, you can observe from the graphic presentation that the data read from the client disk is faster than what can be processed by the network and the TSM server. Consequently, the overall throughput performance may be limited by either the network capacity or the storage device I/O.

If the threads on both client disk and network are similarly busy as shown on the graphic presentation, then the utilization of bandwidth resources is called as balanced, between the client disk and the network. In an optimum setup, the storage device tapes are kept in streaming mode for a balanced use of resource. This means that the network speed may be as fast as the tape I/O speed; for example, there should be no idle time on the network usage.

Generally it is favorable to see a network bottleneck rather than a client disk bottleneck because the tape storage resources may be efficiently used without bottleneck at client disks.

## Performance optimization

By using the Administration Assistant, the performance of TSM environment can be optimized by repeatedly tuning the parameters, modifying the configurations, and verifying the improvements.

The performance optimization cycle starts with a full backup or restore of the data. The data that comes from the performance sensors is analyzed with the function of “View Performance Data,” which may lead to some suggestions on how to change the configurations. The changes are temporarily implemented in a test profile with the function of “Configure Systems.” With the function of “Simulate Backup/Restore,” another backup or restore can be simulated to validate the configuration changes. Then, the function of “View Performance Data” can be re-used to verify whether the modifications yield the desired results. In this way, the cycles of modification and test can be run as many times as required until the final results are satisfactory.

More information about installation and configuration of the Administration Assistant are redirected at References [6].

## 2.5 Using operating system tools

Currently, many operating systems tools are available to monitor system behavior. Some tools can be used effectively to collect the usage data of the CPU and demonstrate the throughput

status of disk, tape, and network I/O, and thus to help isolate, validate, and troubleshoot TSM performance problems. This is especially useful to TSM.

The following are some common tools that are frequently used by the IBM Tivoli Storage Manager:

## FTP

*File Transfer Protocol* (FTP) is used to send files between two systems and is generally available on any system. FTP can be utilized to closely simulate what TSM does during backup and restore of large files through the LAN and can be used to figure out the network bottlenecks.

Typically, FTP throughput is subject to the same considerations as that of TSM, such as the following:

- Source disk read throughput
- LAN throughput
- Target disk write throughput.

FTP can also be configured on some systems to do only network I/O via OS specific tricks.

Note that the best case TSM throughput is typically 50% to 80% of theoretical for TCP/IP using FTP, such as by using the Gb Ethernet, TSM may achieve about 65 - 100 MB/sec and by using the 100Mb Ethernet, TSM may achieve about 6 - 10 MB/sec.

## UNIX dd

*dd* is a command available on UNIX systems to initiate data reads or writes.

By using *dd* together with *time* stamps, data throughput can be calculated practically by transferring a certain amount of data via network I/O, disk I/O or tape I/O in a time period. Therefore, *dd* can be used to explore the behavior of I/O on which the TSM runs.

## Others

There are also many other operating system tools that can be used to help collect performance data or check network segment utilizations.

For validating network status, some network commands such as *ping*, *netstat*, *traceroute* and others are available.

For checking CPU usages and disk behaviors on UNIX, the *iostat* and *vmstat* commands are available. Various flags that are available use the main help page to enable different functionality.

Many tuning commands/tools are platform specific, such as *VMO* and *IOO* for AIX 5, *RMF* for z/OS, *ndd* for Solaris, and *performance monitor / task manager* for Windows.

Moreover, it is effective to learn the tools for each operating system which monitor and simulate the system performance.

---

## 3. Case Study

To cope with real situations on TSM performance, the following basic strategies can be used:

1. Understand the configuration and gather background information.
2. Confirm TSM performance issues by comparing the results to that in other controlled environments or recreate the problem in a controlled environment.
3. Collect TSM operational data with available tools and focus on the operations that take a large amount of time.
4. Analyze the operating system data to identify the highly used system components.
5. Modify how components are used, or add more items such as processors, memory, network adapters, disks, tape drives, and etc.
6. Retest and repeat if necessary to verify TSM performance improvement with the new changes.

Several customer cases are discussed below, with the previous strategies, to examine the typical situations where TSM performance problems are reported, to concentrate on tool usage, and to illustrate the troubleshooting procedures.

- Inefficient disk systems
- Bottleneck at tape systems
- Problematic configurations on TSM server
- File systems with sensitive mount option
- Slow network
- Unbalanced use of resources

For simplifying the explanations, only problem description, problem-solving plan/action, and problem resolution are exhibited for each case study.

### 3.1 Inefficient disk systems

In this case, a problem on disk configuration caused poor TSM disk-to-tape storage pool migration performance and is discovered by using the TSM server instrumentation.

#### **Problem description**

A customer just installed 3592 tape drives on a z/OS server and did not make other changes to the environment. The customer found that TSM disk-to-tape storage pool migration performance was not acceptable with only 7-12 MB/sec per drive, but the tape-to-tape storage pool backup performance was normal with 41 MB/sec per drive.

#### **Problem-solving plan and action**

The customer was requested to rerun a disk-to-tape storage pool migration test in order to collect TSM server instrumentation data.

The "show threads" output shows the pair of migration threads:

```
1390: DfMigrationThread(det), TCB 231, Parent 552, TB 1CD7E18C, N/A 0.
```

1390: Read block disk function (VSAM).

1397: SsAuxThread(det), TCB 159, Parent 1390, TB 1CEDA18C, N/A 0.

1397: WaitCondition WaitBufFull+b8 289C9418 (mutex A89C9328)

TSM server instrumentation shows the performance of migration thread reading the disk as below.

```
Thread 1390 1390 parent=552 15:48:07.050 -->16:20:55.600
Operation      Count  Tottime Avgtime Mintime Maxtime InstTput  Total KB
-----
-
Disk Read      58852 1833.997  0.031  0.000  0.512   8213.8 15064160
Acquire Latch 19126   0.091  0.000  0.000  0.002
Thread Wait   58858  89.360  0.002  0.000  60.917
Unknown                          45.100
-----
Total                          1968.550                               7652.4 15064160
```

It clearly shows that Disk Read takes 1833.997 seconds out of 1968.550 seconds in total migration time. A large amount (93% =  $100 \times 1833.997 / 1968.550$ ) of time in Disk Read indicates that that disk is the bottleneck.

Note that by dividing the total KB read by the count, the read I/O size is shown as 256 KB (=  $15064160 \text{ KB} / 58852 \text{ Counts}$ ), which is normal.

### Problem resolution

This customer was advised to create new disk storage pool volumes by using a newer disk subsystem and VSAM striped datasets. Then, disk -to-tape storage pool migration performance returned back to normal: 37 - 40 MB/sec per drive.

## 3.2 Bottleneck at tape system

In this case, a performance bottleneck on the tape system is identified by using the TSM server instrumentation.

### Problem description

The customer has a Windows server with SCSI -attached LTO1 tape drives. The Microsoft Exchange clients are located on AIX and Windows platforms. The customer reported that for all clients, the backup to tape is slow, such as backup 30 GB Exchange data took 7 hours, or 1.2 MB/sec. Also, disk-to-tape storage pool migration performance was slow.

### Problem-solving plan and action

The customer was asked to rerun a disk-to-tape storage pool migration test for collecting server instrumentation data.

TSM server instrumentation shows the performance of the migration threads as below:

```
Thread 61 DfMigrationThread (Win Thread ID 4436) 17:39:076-->17:47:38
Operation      Count  Tottime  Avgtime  Min -   Max-   Inst   Total
              time   time     time    time   time   Tput   KB
-----
Disk Read      3777   22.680   0,006   0.000  0.031  42632,8 966912
Thread Wait    3778   487.450  0,129   0.016  0.313
Unknown                               0.061
-----
Total                               510.191                1895,2 966912
```

```
Thread 34 AgentThread (Win Thread ID 5340) 17:39:07.816-->17:47:38.007
Operation      Count  Tottime  Avgtime  Min -   Max-   Inst   Total
              time   time     time    time   time   Tput   KB
-----
Tape Write     30257   508.816  0,017   0.000  0.141  1902,8 968192
Tape Data Copy 31661   0.863   0,000   0.000  0.016
Thread Wait    3777   0.220   0,000   0.000  0.016
Unknown                               0.292
-----
Total                               510.191                1897.7 968192
```

It shows that Thread Wait takes 487.450 seconds out of 510.191 seconds in total time of dat a migration and Tape Write takes 508.816 seconds out of 510.191 seconds in total time of write data to tape.

A large amount of time used on Thread Wait and Tape Write indicates that the problem is clearly related to the tape system. Further investigation is needed to focus on the following components on the tape system:

- Tape attachment path
- Tape drive device driver level
- SCSI adapter driver level
- SCSI adapter settings

### **Problem resolution**

Customer upgraded the SCSI adapter device driver.

Presently, disk-to-tape storage pool migration performance is good with normal 19 MB/sec per drive and client backups to tape are much faster too.

## **3.3 Problematic configuration of TSM server**

In this case, a TSM server configuration issue that impacts backup performance is tracked out by using TSM client instrumentation.

### **Problem description**

This AIX server is reported to have a slow incremental backup of Windows' clustered file server. TSM backed up 25 GB of data using 39 hours that is the throughput of only 187 KB/sec.

The client session report is shown as below.

```
Session established with server TSM_WIN_DL_1: AIX -RS/6000

Server Version 5, Release 2, Level 3.0

Total number of objects inspected:    3.268.763
Total number of objects backed up:    485.742
.....
Total number of bytes transferred:    25,44 GB
Network data transfer rate:           5.609,21 KB/sec
Aggregate data transfer rate:         188,34 KB/sec
Elapsed processing time:               39:20:39
Average file size:                     54,15 KB
```

### **Problem-solving plan and action**

The customer was asked to rerun the backup test for collecting client instrumentation data.

The TSM client instrumentation shows the performance of the backup threads below:

Thread: 10856 Elapsed time 141638,718 sec

Section	Actual (sec)	Average(msec)	Frequency used
Compute	15,993	0,0	1288373
BeginTxn Verb	0,294	0,0	10109
Transaction	2082,912	206,0	10109
File I/O	7697,765	4,2	1813428
Data Verb	4747,407	3,7	1288373
Confirm Verb	2,376	81,9	29
EndTxn Verb	49562,039	4902,8	10109
Other	77529,932	0,0	0

Thread: 11048 Elapsed time 141618,671 sec

Section	Actual (sec)	Average(msec)	Frequency used
Process Dirs	30891,674	98,3	314384
Other	110726,997	0,0	0

Noticed was that EndTxn Verb takes 49562.039 seconds, which is much longer than that of other operations except "Other".

A large amount of time was spent in EndTxn, in comparison with other TSM operations. This indicates a problem with the TSM server database or tape storage.

Further investigation on the TSM server configuration uncovered that the TSM server database, recovery log, and disk storage pool volumes were configured on the same LUN.

### Problem resolution

The customer was guided to reconfigure the TSM server database separately and use multiple volumes on multiple LUNs. The next incremental backup took 17 hours (2 times faster).

Additional network configuration problems were found and fixed, and the customer implemented using journal-based incremental backup, which now takes about 4 hours (10 times faster) for backup of the same data.

## 3.4 File systems with sensitive mount option

In this case, a poor disk performance issue due to the mount option cio for AIX concurrent I/O was discovered by using TSM client instrumentation.

## Problem description

A customer used an AIX server and reported that the performance of the backup Oracle database on an AIX client was degraded from 32 MB/sec to 16 MB/sec after rebooting the client machine on which TSM backup/archive client is also used. The client session report is shown below:

```
Total number of objects inspected:          1
Total number of objects backed up:          1
.....
Total number of bytes transferred:          11.80 GB
LanFree data bytes:                        11.80 GB
Server-Free data bytes:                    0 B
Data transfer time:                        216.01 sec
Network data transfer rate:                57,294.91 KB/sec
Aggregate data transfer rate:              16,542.69 KB/sec
Elapsed processing time:                   00:12:28
Average file size:                         11.66 GB
```

## Problem-solving plan and action

The customer was asked to rerun the backup test to collect client instrumentation data.

TSM client instrumentation shows the performance of the backup thread below:

```
Thread: 2571 Elapsed time 746.666 sec
Section      Actual (sec)      Average(msec)  Frequency used
-----
Process Dirs    0.000           0.0            0
Solve Tree     0.000           0.0            0
Compute        0.234           0.0           48345
BeginTxn Verb  0.000           0.1            2
Transaction    0.715          357.5           2
File I/O       524.380         10.8           48346
Compression    0.000           0.0            0
Encryption     0.000           0.0            0
CRC            128.042         2.6           48398
Delta          0.000           0.0            0
Data Verb      87.912          1.8           48345
```



Confirm Verb	0.136	8.5	16
EndTxn Verb	2.234	1117.0	2
Other	4.513	0.0	0

---

It shows that File I/O takes 524.380 seconds out of 746.666 seconds in total elapsed time.

A large amount of time used in File I/O indicates a problem with reading the client file data. A simple calculation shows that the File I/O throughput is only about 23 MB/sec (= Total data transferred 11.80 x 1024 MB / Actual File I/O time 524.380 sec).

The customer mentioned that the file system had recently been changed to be mounted with the cio option, which can enable AIX Concurrent I/O.

It is known that using AIX concurrent I/O as a file system mount option will have no file system read ahead, and cause the backup throughput degradation

### Problem resolution

The customer was recommended to install TSM 5.3 client that allows setting the file read I/O size by means of the TSM client diskbuffsize option, and diskbuffsize 1020 is used. The backup throughput improved to 80% of the previous performance.

Note that some databases (such as DB2) can use CIO on the file open call. But this file system mount option is not necessary to be used, and there is no impact on backup performance.

## 3.5 Slow network

In this case, a slow network issue that caused poor TSM backup performance is isolated by using TSM server instrumentation and validated by using FTP.

### Problem description

An AIX server is attached with LTO 1 and 3590 tape drives. The customer reported that Exchange client backup to tape is slow with the throughput of only 240 KB/sec.

### Problem-solving plan and action

The customer was asked to rerun a backup test to collect TSM server instrumentation data. Note that this is an API client, so it is not feasible to take TSM client instrumentation data.

The TSM server instrumentation shows the performance of the backup threads below:

```
Thread 37 SessionThread parent=32 (AIX TID 47241) 15:55:39 -->16:11:04
Operation      Count   Tottime  Avgtime   Min Maxtime  InstTput  Total KB
-----
Network Recv   137432  762.793  0.006 0.000  11.234    301.9    230298
Network Send    42      0.006   0.000 0.000   0.000    1055.3     7
Acquire Latch   84      0.000   0.000 0.000   0.000
Thread Wait    900     21.047  0.023 0.000  18.858
```

Unknown		141.200					
-----							
Total		925.048				249.0	230305
Thread 58 <i>AgentThread</i> parent=55 (AIX TID 47681) 15:55:41 -->16:11:04							
Operation	Count	Tottime	Avgtime	Min	Maxtime	InstTput	Total KB
-----							
Tape Read	4	0.423	0.106	0.000	0.364	0.0	0
Tape Write	905	14.695	0.016	0.000	0.691	15661.1	230144
Tape Locate	1	0.007	0.008	0.000	0.007		
Tape Data Copy	1585	2.690	0.002	0.000	0.003		
Acquire Latch	45	0.000	0.000	0.000	0.000		
Thread Wait	904	884.445	0.978	0.000	12.402		
Unknown		21.520					
-----							
Total		923.783				249.1	230144

You can see that the network receiving time (Network Recv) takes 762.793 seconds out of 925.048 seconds in the total time of receiving data from the client.

A large amount of time was spent on the network receiving, indicating that the problem is clearly related to receiving data from the client. It could be either a network or a client problem.

The customer was then advised to measure FTP throughput on the same network route. The FTP results show that this network only can achieve the throughput of 392.85 KB/sec:

---

```

ftp> put pippo.log
200 PORT command successful.
150 Opening data connection for pippo.log.
226 Transfer complete.
ftp: 797290032 bytes sent in 2029.48Seconds 392.85Kbytes/sec.
ftp>

```

---

The output indicates a clear network problem, and TSM should never get any better throughput on this network.

### Problem resolution

The customer was asked to reconfigure the network, which improved TSM throughput performance.

### 3.6 Unbalanced use of resources

In this case, with help from the Administration Assistant, storage resource usage is optimized and the overall throughput is increased for a database backup using TSM for ERP.

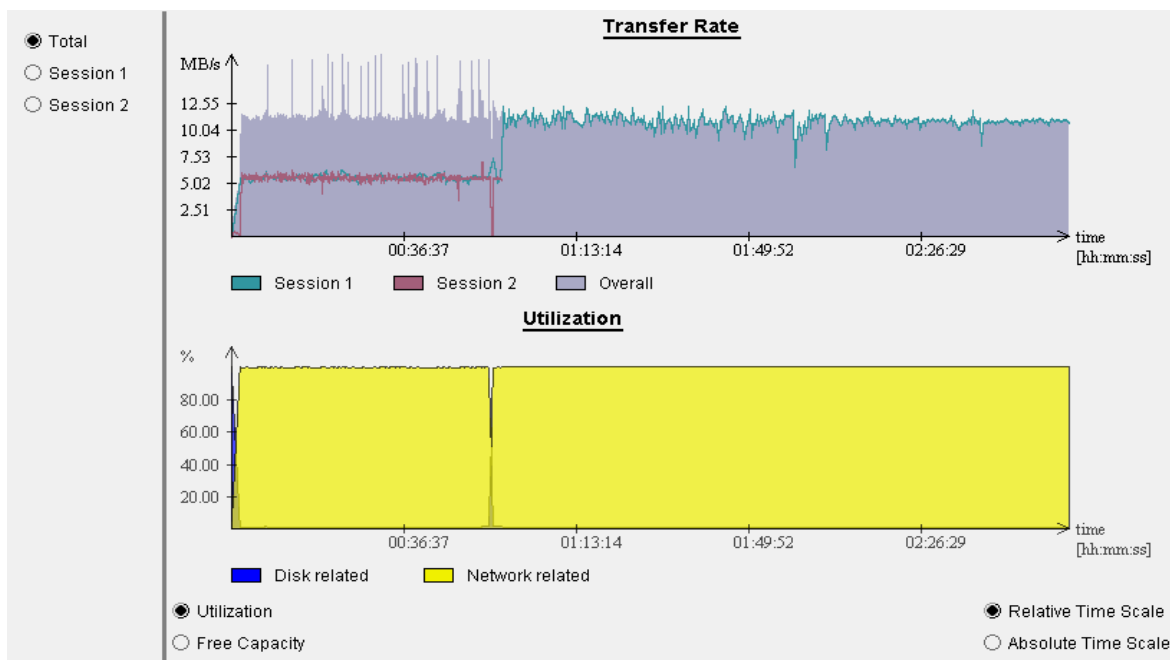
#### Problem description

A customer maintained a database of approximately 115 GB. TSM for ERP was employed to back up the database, and two tapes were used, with multiplexing four data streams to each tape. With this setup, the backup throughput of approximately 38 GB/hour, or 10.8 MB/sec, is achieved. Is this configuration optimized?

#### Problem-solving plan and action

The customer was asked to perform a TSM backup with the Administration Assistant enabled to collect system performance data and observe resource utilizations.

The Administration Assistant's graphic presentation in figure 1 shows the output created by analyzing the data transfer rate and resource utilization.



**Figure 1: Data Transfer Rate and Resource Utilization before modifications**

The *Utilization* chart shows an unbalanced bandwidth usage between disk and network, and the bottleneck is network related (denoted by yellow). This indicates that the overall throughput is limited by either network or tape storage devices.

The *Transfer Rate* chart for session 2, denoted by the red line, ends much earlier than session 1, which is denoted by the green line. This is due to the fact that the files that are sent via session 1 to the first tape are much larger than the ones processed in session 2 to the second tape. However, the overall throughput rate denoted by the grey area does not drop significantly when session 2 ends. This indicates that the network capacity limits the throughput rather than the tape

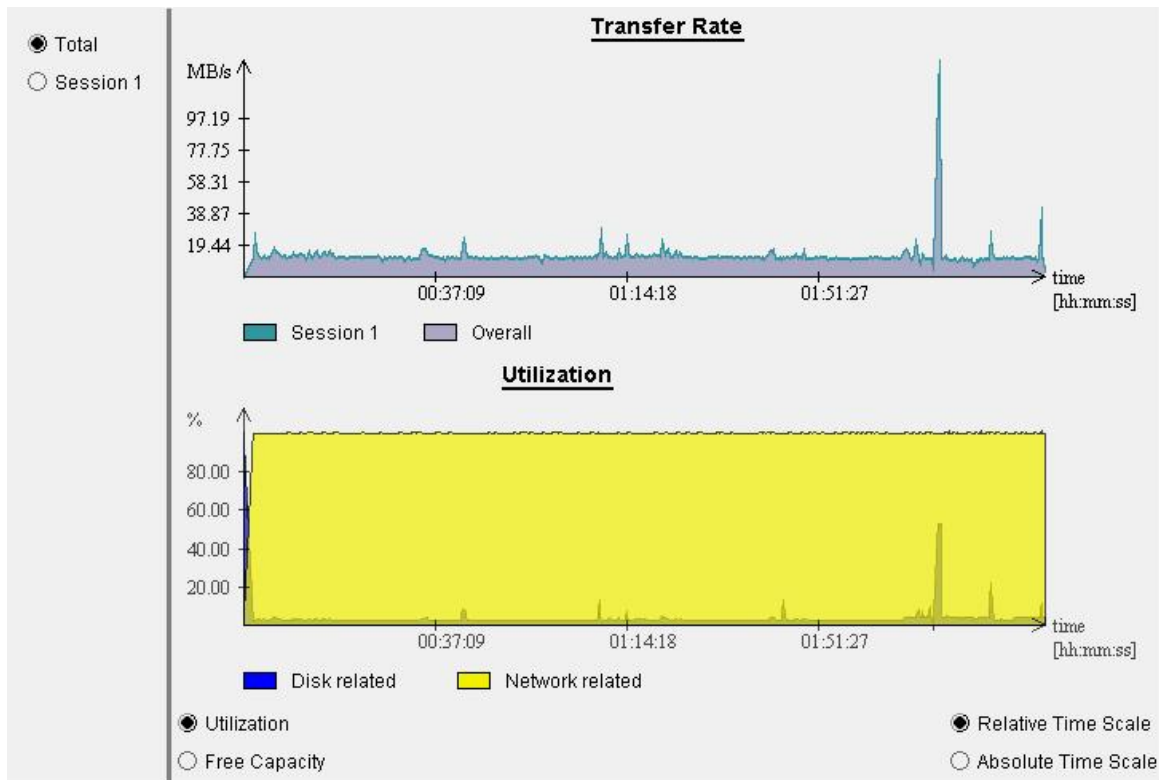
capacity. As a consequence, we can conclude that two tapes may not be used efficiently in this environment.

### Problem resolution

The customer was asked to do a backup using a single tape with a multiplexing level of 2 instead of 4 (Thus, two client disks can still be written to simultaneously during restore).

To further relieve the strain on the network, the customer was asked to use the *compression* option provided by TSM for ERP.

With these modifications, one of the original two tapes was dropped so that the storage resource (tape devices) was used more efficiently. The overall throughput was increased to approximately 46 GB/hour or 13 MB/sec, which is an increase of about 20 %, while using less storage resource. See figure 2 below:



**Figure 2: Data Transfer Rate and Resource Utilization after modifications**

Note that in figure 2, the scale of the graphic presentation showing the transfer rate is changed because of a peak due to a file with an extreme compression rate. This coincides with a higher utilization of the client disks at this point in time.

Restore tests showed that the restore performance was also improved with the new configuration.

---

## 4. Summary

It is challenging to find the performance bottlenecks in a complicated TSM environment, as there are so many factors that may affect TSM performance across the components of client/server, network, and storage devices.

In this paper, the importance of TSM configuration with performance perspectives is emphasized, the recommended parameters and basic clues for TSM performance tuning are provided, and the handy tools for TSM performance troubleshooting are demonstrated to help users face the challenge.

Generally, a list of questions, as below, could be summarized to explain the issues of TSM performance configuration:

- Is the server database optimized?
- Is the server device I/O optimized?
- Is data collocation used?
- Is transaction size increased?
- Are the network options tuned?
- Is LAN-Free used?
- Is incremental backup used?
- Is multiple client sessions used?
- Is image backup/restore used?
- Is scheduler optimized?

If you follow this list appropriately and the questions are answered completely, perhaps you may not have any performance problems.

Specifically, during various TSM operations, the common performance problems can be diagnosed with the following actions:

For poor **backup to tape** performance, it is essential to verify if there is: 1) High tape mount wait time; 2) Poor client disk read performance; 3) Poor network performance; or 4) Small TSM client transaction size.

For poor **backup to disk** performance, it is basic to examine if there is: 1) Poor network performance; 2) Contention with other backup/archive sessions or other processes; 3) Poor client disk read performance; or 4) Poor TSM server database performance.

For poor **inventory expiration** performance, it is reasonable to explore if there is: 1) Poor TSM server database performance; 2) Contention with backup/archive sessions or other processes; or 3) Slow TSM server CPU.

For poor **restore from tape** performance, it is usual to check if there is: 1) High tape mount wait time; 2) Large number of tape mounts or locates; 3) Poor network performance; 4) Poor client disk write performance; 5) Poor TSM server database performance; or 6) Data collocation.

For poor **storage pool migration** performance, it is important to confirm if there is: 1) High tape mount wait time; 2) Large number of tape mounts; 3) Poor TSM server disk

read performance; 4) Contention with backup/archive sessions or other processes; 5) Close migration thresholds setup (the thresholds may be set up too close together); or 6) Small TSM server data movement transaction size.

TSM performance facilities, including TSM instrumentation, Administration Assistant, and some operating system tools, are documented in this paper. It is convenient to use the facilities for collecting test data and then help narrow down the large TSM operational times onto the costly components of server, client, network, and storage device. Thus, together with problem -solving strategies and fundamental approaches demonstrated through effective troubleshooting, the real customer situations and the corresponding resolutions can be properly generated to resolve TSM performance problems.

In summary, no matter how complicated the situation might be, there may be a bottleneck in the TSM environment. The bottleneck identification, or problem isolation, is a critical step towards an optimal resolution which can satisfy the expectations of our customers and be accepted by the performance professionals of IBM Tivoli Storage Manager.

---

## 5. References

- [1] For more information about TSM performance tuning and troubleshooting, visit the Tivoli Information Center where the TSM Performance Tuning Guide, the TSM Problem Determination Guide, and many other TSM Documentations are located:  
<http://publib.boulder.ibm.com/infocenter/tivihelp/v1r1/index.jsp>
- [2] For the performance information of TSM Administration Center, visit *TSM 532 Administration Center Performance Evaluation Report* at the following link:  
<http://www-1.ibm.com/support/docview.wss?uid=swg21193443&aid=1>
- [3] The online presentation of *Troubleshooting Performance Bottlenecks in Tivoli Storage Manager Environment*, as the course of *TSM303t* in IBM Tivoli Distance Learning program, is downloadable from this Web site:  
[http://publib.boulder.ibm.com/tividd/software/saleskits/B591078S12403L99/KEY\\_19.html](http://publib.boulder.ibm.com/tividd/software/saleskits/B591078S12403L99/KEY_19.html)
- [4] The presentation of *Finding Performance Bottlenecks in TSM Environment*, as the session 2897 in IBM Software University 2006, is downloadable from this Web site:  
[http://w3-03.ibm.com/software/sales/salesite.nsf/salestools/Technical+Sales\\$SU2006+Tivoli+recordings](http://w3-03.ibm.com/software/sales/salesite.nsf/salestools/Technical+Sales$SU2006+Tivoli+recordings)
- [5] To follow the procedure of configuring multiple TSM servers on one machine, search the key words of "Multiple TSM servers" in the following Web site:  
<http://www-1.ibm.com/support/>
- [6] The detailed information about Administration Assistant is covered in the Chapter 9 (for installation) and Chapter 10 (for usages) of the *IBM Tivoli Storage Manager for Enterprise Resource Planning, Data Protection for mySAP, Installation and User's Guide for Oracle*, which is located in the following Web site:  
<http://publib.boulder.ibm.com/infocenter/tivihelp/v1r1/index.jsp?toc=/com.ibm.itsmerp.doc/toc.xml>
- [7] For further information of Tivoli Storage Manager Products Technical Support, see the following Web site:  
<http://www-3.ibm.com/software/sysmgmt/products/support/IBMTivoliStorageManager.html>
- [8] For general Tivoli Storage Manager product information, the Tivoli Storage Manager Home Page is the best place to surf:  
[http://www-306.ibm.com/software/tivoli/products/storage\\_mgr/](http://www-306.ibm.com/software/tivoli/products/storage_mgr/)
- [9] For more information on TSM incremental backup memory requirements, refer to:  
<http://www-1.ibm.com/support/docview.wss?rs=663&uid=swg21197422>