Special Notices

Disclaimer

The performance data contained in this presentation was measured in a controlled environment. Results obtained in other operating environments may vary significantly depending on factors such as system workload and configuration. Accordingly, this data does not constitute a performance guarantee or warranty.

References in this presentation 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 programs may be used. Any functionally equivalent program may be used instead.

Trademarks and Registered Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

AIX
IBM
Tivoli
TotalStorage
z/OS

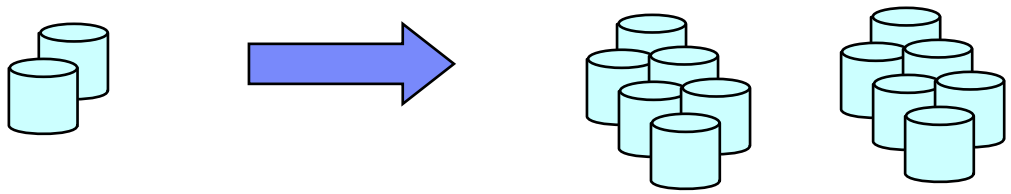
Other company, product, and service names may be trademarks or service marks of others.

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.

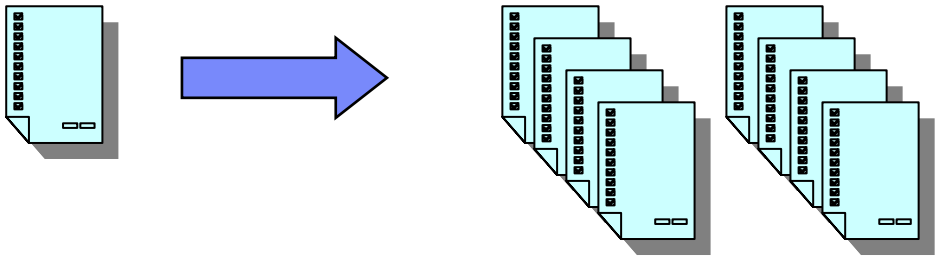
1. Optimize Server Database

- **TSM server database performance is important for many operations**
- **Use multiple database volumes (4 to 16)**
- **Use a separate disk (LUN) for each database volume**
- **Use fast disks (high RPM, low seek time, low latency)**
- **Use disk subsystem/adapter write cache appropriately**
  - Only use protected write cache (must be battery-backed, NVS, ...)
  - Use for all RAID arrays
  - Use for all physical disks with TSM database volumes
  - Do not use for physical disks with TSM storage pool volumes
- **Use *dbpageshadow yes* server option**
  - Page shadow file can be placed in the server install directory
- **Use *mirrorwrite db parallel* when using volumes mirrored by TSM**



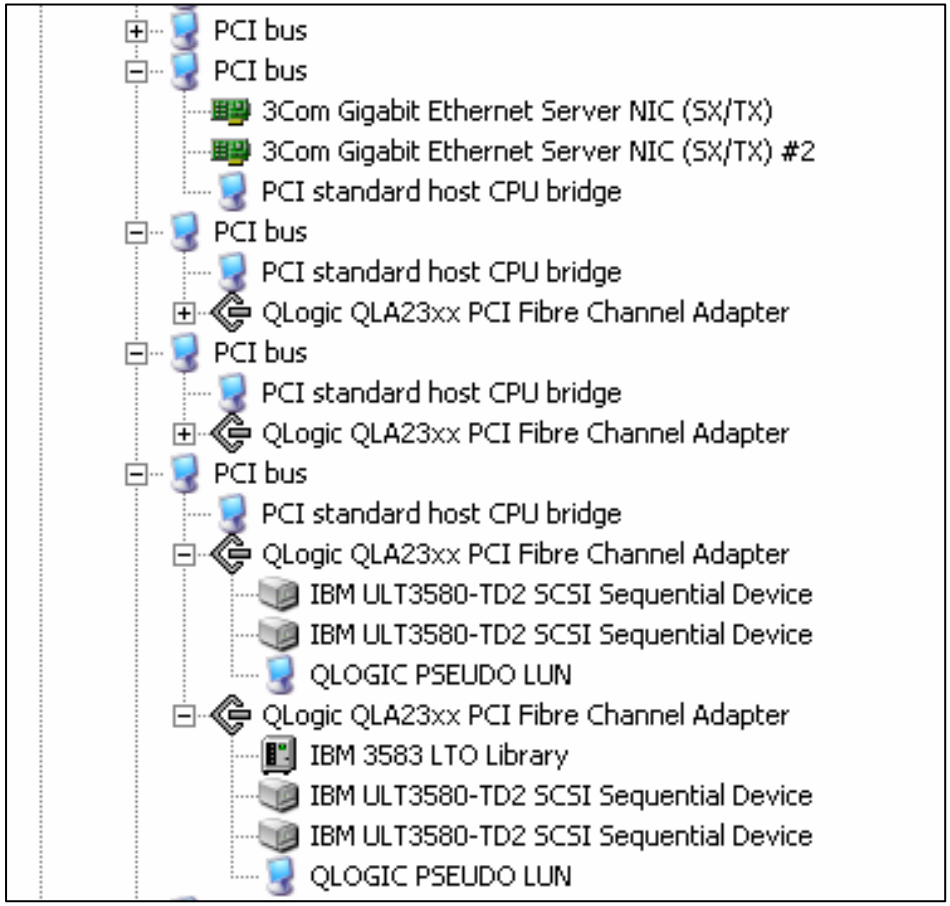
1. Optimize Server Database (cont.)

- **Increase the size of the database buffer pool**
  - Server option ***bufpoolsize*** is specified in KB
  - TSM 5.3 default is 32768 (32MB)
  - Initially set to 1/4 of server real memory or process virtual memory limit (whichever is lower)
    - ✓ Example: 32bit Server has 2 GB RAM; set ***bufpoolsize 524288***
  - DO NOT increase the buffer pool if system paging is significant
  - Use the **query db** command to display the database cache hit ratio
    - ✓ TSM server cache hit ratio should be greater than 98%



2. Optimize Server Device I/O

- **Server performance depends on the system I/O throughput capacity**
- **Study system documentation to learn which slots use which PCI bus**
  - Put fastest adapters on the fastest busses
- **For best LAN backup-to-disk performance:**
  - Put network adapters on different bus than disk adapters
- **For best disk-to-tape storage pool migration performance:**
  - Put disk adapters on different bus than tape adapters

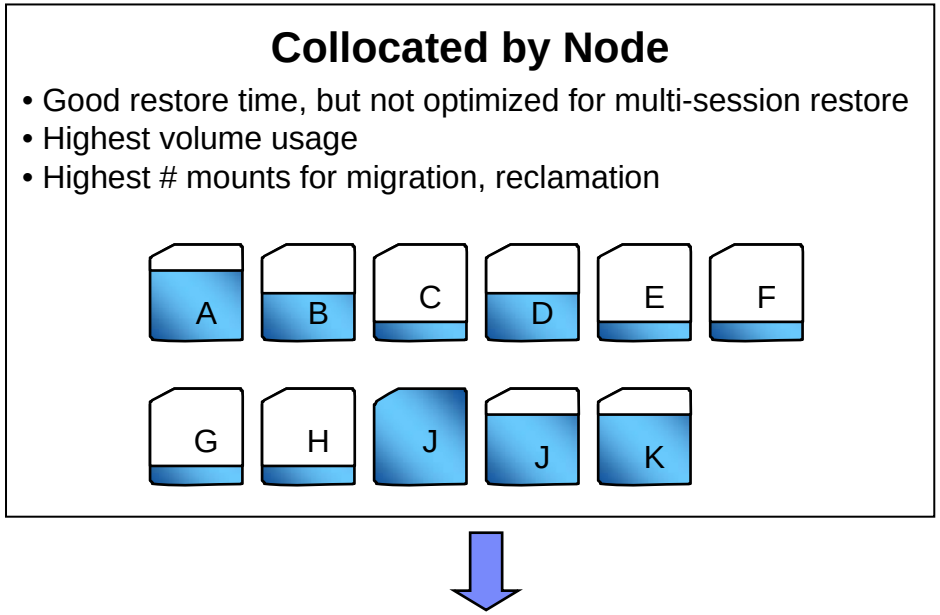


2. Optimize Server Device I/O (cont.)

- **Parallelism allows multiple concurrent operations**
- **Use multiple:**
  - Busses
  - Adapters
  - LANs and SANs
  - Disk subsystems
  - Disks
  - Tape drives
- **If using a DISK storage pool (random access)**
  - Define multiple disk volumes
  - One volume per disk (LUN)
- **If using a FILE storage pool (sequential access)**
  - Use multiple directories in the device class (new in TSM 5.3)
  - One directory per disk (LUN)
- **Configure LUNs within a disk subsystem with regard for performance**
- **Configure enough disk storage for at least one day's backups**
- **Configure at least as many disk volumes or directories as storage pool migration processes**

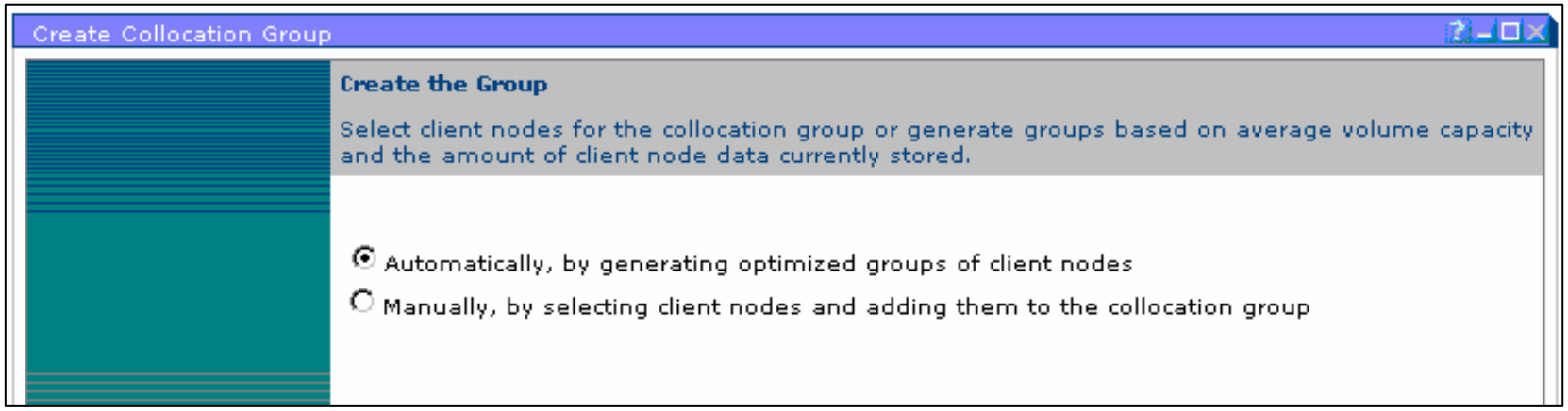
3. Use Collocation by Group

- **New in TSM 5.3 !**
- **For sequential-access storage pools**
- **Note: Use collocation by filespace for nodes with 2 or more large filespaces**



3. Use Collocation by Group (cont.)

- **Must define the collocation groups and their nodes**
- **Manual methods:**
  - Administration Center *Create Collocation Group* panel
  - **define collocgroup**, **define collocmember**, etc. server commands
- **Automatic methods:**
  - Administration Center *Create Collocation Group* panel
  - **defgroups** sample PERL script in the “server” directory or SAMPLIB



3. Use Collocation by Group (cont.)

- **Automatic methods:**
  - Create new collocation groups to include all non-grouped nodes
  - Specify the domain and a collocation group name prefix
  - Group size based on volume capacity and node occupancy
    - ✓ Administration Center wizard uses 4 times the average full volume capacity
    - ✓ **defgroups** requires capacity to be specified (use **fullvolcapacity** to determine)

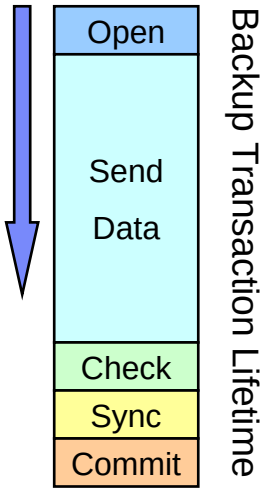
```
#defgroups -id=canichol -pass=********** standard devgroup 60000

                                 Number of nodes Number of MBs of
                                 defined to      physical space used
Group name                       this group      by these nodes
-------------------------------- --------------- -------------------
DEVGROUP1                                      1              706073
DEVGROUP2                                      1              101441
DEVGROUP3                                      1               82219
DEVGROUP4                                      3               59920
DEVGROUP5                                      3               59800
DEVGROUP6                                      4               59954
DEVGROUP7                                      4               59916
DEVGROUP8                                      4               59835
...
```

3. Use Collocation by Group (cont.)

- **Automatic methods:**
  - Don't know about relationships between nodes
  - Don't know about node data growth rates
  - Don't know how existing data is collocated
- **Suggestions:**
  - Use the automatic methods first and then fine tune
  - Group nodes together that have a low chance of restore at the same time
    - ✓ Avoid volume contention
  - Group nodes together that backup to disk at the same time
    - ✓ If storage pool migration has to run during the backup window
    - ✓ Reduced volume mounts since all data for a group can be moved at one time
- **Data is moved when storage pool migration, reclamation, etc. run**
  - Using the collocation group and storage pool definitions at that time
- **Optimal data organization should occur over time**

4. Increase Transaction Size

- **Increase backup and server data movement throughput**
- **Set the following server options (TSM 5.3 defaults):**
  - ***txngroupmax 256***
  - ***movebatchsize 1000***
  - ***movesizethresh 2048***
- **Set the following backup/archive client option:**
  - ***txnbytelimit 25600***
- **For nodes that back up small files direct to tape:**
  - Update the node definition:
    - **update node *nodename* txngroupmax=4096**
  - Set the following client option:
    - ***txnbytelimit 2097152***

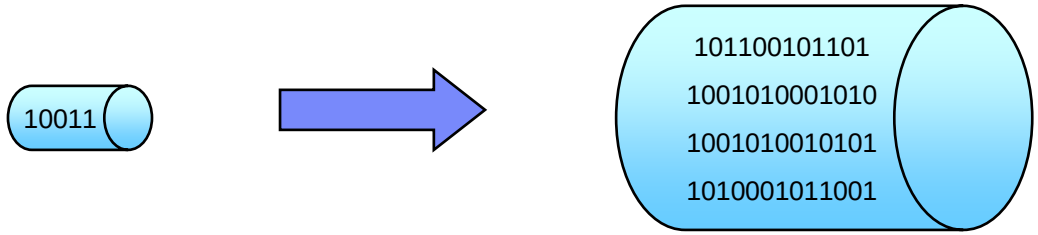


4. Increase Transaction Size (cont.)

- **May need to increase the TSM server recovery log size**
  - 4 GB is sufficient for many installations (13.5 GB max)
- **Backup throughput could degrade if frequent file retries**
  - A retry occurs when processing is interrupted
  - Throughput is degraded because the data must be sent again
  - Check the client session messages or schedule log file (***verbose***)
  - Avoid retries by:
    - Using client option ***compressalways yes***
    - Using client option ***tapeprompt no*** or ***quiet***
    - Scheduling backup / archive when files are not in use
    - Using exclude options to exclude files likely to be open
    - Using Open File Support (Windows client)
    - Changing how open files are handled using the copy group ***serialization*** parameter
    - Reducing the number of times to retry using client option ***changingretries***

5. Configure and Tune Network Options

- **Provide a separate backup network (LAN or SAN)**
  - Gb Ethernet can provide up to 75 MB/sec throughput per link
  - 100 Mb Ethernet can provide up to 11 MB/sec throughput per link
- **Set the following server options (TSM 5.3 defaults):**
  - New options for the TSM 5.3 z/OS server!
  - ***tcpwindowsize 63***
  - ***tcpnodelay yes***
  - ***tcpbufsize 32*** (not used for Windows)
- **Set the following client options:**
  - ***tcpwindowsize 63***
  - ***tcpnodelay yes***
  - ***tcpbuffsize 32***
  - ***largecommbuffers no*** (replaced in TSM 5.3)



5. Configure and Tune Network Options (cont.)

- **Use TSM client compression appropriately**
- **Software compression uses significant client CPU time!**
- **Set the following client options:**
  - ***compression***
    - ✓ Fast network AND fast server - ***compression no***
    - ✓ For LAN-Free with tape - ***compression no***
    - ✓ Slow network OR slow server - ***compression yes***
  - ***compressalways yes***
- **Exclude objects that are already compressed or encrypted:**
  - ***exclude.compression ?:\...*.gif***
  - ***exclude.compression ?:\...*.jpg***
  - ***exclude.compression ?:\...*.zip***
  - ***exclude.compression ?:\...*.mp3***

6. Use LAN-Free

- **Offload the LAN**
- **Better TSM server scalability due to reduced I/O requirements**
- **Higher backup/restore throughput is possible**
- **Best with "TSM for …" data protection products and API clients that backup/restore big objects**
  - TSM for Mail
  - TSM for Databases
  - TSM for Enterprise Resource Planning
- **Performance improvements in TSM 5.2:**
  - Reduced storage agent - server meta-data overhead
  - Better multi-session scalability
  - Better storage agent tape volume handling
- **Performance improvements in TSM 5.3:**
  - Reduced CPU usage, especially for API clients
  - Use ***lanfreecommmethod sharedmem*** for all TSM 5.3 LAN-Free clients
    - ✓ AIX, HP-UX, HP-UX on Itanium2, Linux, Solaris, and Windows

7. Use Incremental Backup

- **Weekly full backups for file servers aren't necessary**
- **TSM incremental backup**
  - Compares client file system with server inventory
  - Backs up new or changed files and directories
  - Expires deleted files and directories
  - No unnecessary data backed up
  - Less network and server bandwidth needed
  - Less server storage needed
- **Windows file servers should use Journal-based incremental backup**
  - Install using the client GUI Setup wizard
  - Real-time determination of changed files and directories
  - Avoids file system scan and attribute comparison
  - Much faster than full incremental
  - New support in TSM 5.3
    - ✓ More reliable journal database
    - ✓ Multi-threaded operation on both change notification and back up operations
    - ✓ Faster and more reliable multiple file system backup

8. Use Multiple Client Sessions

- **Multiple parallel sessions can improve throughput for backup and restore**
- **Configure multiple sessions for "TSM for …" data protection products**
  - Each has its own configuration options
- **Configure multiple sessions for the Backup/Archive client**
  - Set client option ***resourceutilization*** to 5 or more
- **Update the maximum number of mount points for the node**
  - For backup or restore using tape
  - **update node … maxnummp=4** server command
- **UNIX file servers should use the TSM client option *virtualmountpoint* to allow multiple parallel incremental backups for large file systems**
- **Multi-session restore is only used if:**
  - Restore specification contains an unqualified wildcard, i.e. e:\users*
  - Restore data is stored on multiple sequential storage pool volumes

9. Use Image Backup

- **Optimizes large file system restore performance**
  - Uses sequential block I/O
  - Avoids file system overheads, including file open(), close(), etc.
- **Throughput can approach hardware device speeds**
- **Online image backup is available:**
  - Windows, Linux86, and Linux IA64 clients
    - ✓ Uses LVSA snapshot agent
    - ✓ LVSA cache can be located on a volume being backed up (for TSM 5.3)
- **Recommendations:**
  - Use LAN-Free with tape for best performance
  - Use parallel image backup/restore sessions for clients with multiple file systems

10. Optimize Schedules

- **Create schedules with minimal overlap**
- **Reduce resource contention and improve performance**
- **Operations:**
  - Client backup
  - Storage pool backup
  - Storage pool migration
  - TSM database backup
  - Inventory expiration
  - Reclamation
- **Use set randomize *percent***
  - Client session start times are staggered over the schedule window
- **Use server option *expinterval 0* to disable automatic expiration**
  - Define an administrative schedule for expiration at a set time
- **TSM 5.3 includes scheduling improvements**
  - Calendar-type administrative and client schedules
  - New commands include **migrate stgpool**, **reclaim stgpool**

Summary Checklist

- Optimized the server database?
- Optimized the server device I/O?
- Using collocation by group?
- Increased transaction size?
- Tuned network options?
- Using LAN-Free?
- Using incremental backup?
- Using multiple client sessions?
- Using image backup/restore?
- Optimized schedules?


IBM
Tivoli Storage, IBM Software Group
Finding Performance
Bottlenecks in your
Tivoli Storage Manager Environment
IBM TotalStorage Open Software Family
© 2005 IBM Corporation

Performance Steps:

1. **Use the Top 10 Performance Checklist**
2. **Use IBM Support**
3. **Check for common problems**
4. **Find the bottleneck - determine the limiting factors**
5. **Tune or make configuration changes**
6. **Reduce the workload by eliminating non-essential tasks**
7. **Install and configure additional hardware**
8. **Collect and use performance trend data**

1. Top 10 Performance Checklist

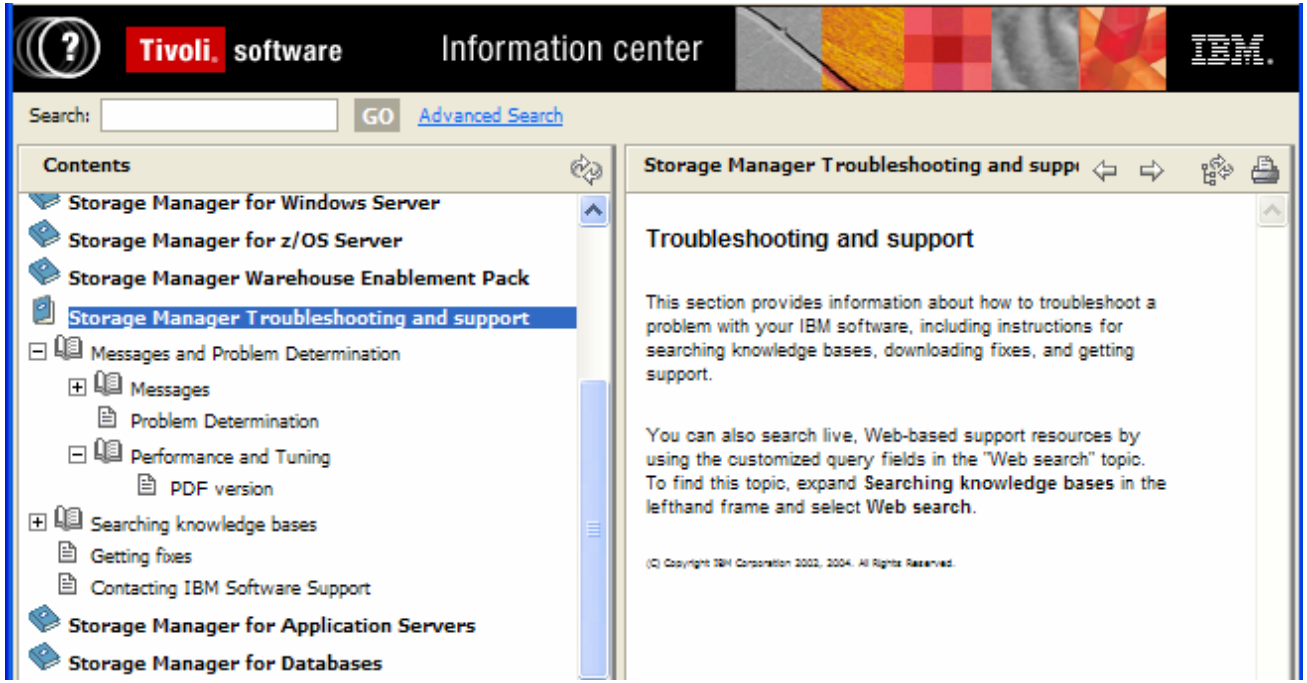
- Optimized the server database?
- Optimized the server device I/O?
- Using collocation by group?
- Increased transaction size?
- Tuned network options?
- Using LAN-Free?
- Using incremental backup?
- Using multiple client sessions?
- Using image backup/restore?
- Optimized schedules?

2. Use IBM Support

http://publib.boulder.ibm.com/infocenter/tivihelp/index.jsp?toc=/com.ibm.itstorage.doc/toc.xml

Look under Storage Manager Troubleshooting and Support

- Knowledge base search
- TSM Performance Tuning Guide

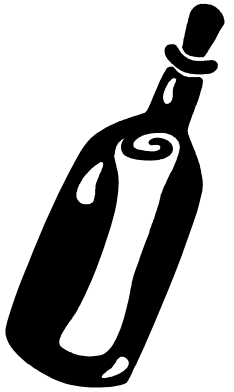


3. Check for Common Problems

- **Poor backup to tape performance?**
  - High tape mount wait time?
  - Poor client disk read performance?
  - Poor network performance?
  - Small TSM client transaction size?
- **Poor backup to disk performance?**
  - Poor network performance?
  - Contention with other backup/archive sessions or other processes?
  - Poor client disk read performance?
  - Poor TSM server database performance?
  - Incremental backup?
- **Poor inventory expiration performance?**
  - Poor TSM server database performance?
  - Contention with backup/archive sessions or other processes?
  - Slow TSM server CPU?
- **Poor restore from tape performance?**
  - High tape mount wait time?
  - Large number of tape mounts or locates?
  - Poor network performance?
  - Poor client disk write performance?
  - Poor TSM server database performance?
- **Poor storage pool migration performance?**
  - High tape mount wait time?
  - Large number of tape mounts?
  - Poor TSM server disk read performance?
  - Contention with backup/archive sessions or other processes?
  - Migration thresholds set too close together?
  - Small TSM server data movement transaction size?

4. Find the Bottleneck

1. **Collect TSM instrumentation data to isolate the cause to:**
   - Client
   - Server
   - Network
2. **Collect operating system data to find the high use component**
3. **Change how that component is used, or add more of it**
   - Processors, memory, disks, tape drives, ...
4. **Retest and repeat if necessary**
   - There is always a bottleneck!

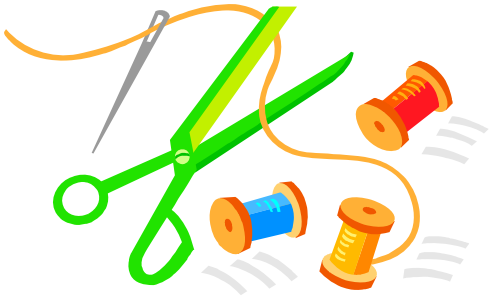


TSM Performance Instrumentation

- **Available on both server and client**
- **Minimal performance impact**
  - No huge trace file created
- **What is tracked?**
  - Most operations that can hold up performance
    - ✓ Disk I/O
    - ✓ Network I/O
    - ✓ Tape I/O
- **How is it tracked?**
  - Operations are tracked on a thread-by-thread basis
  - Most sessions/processes use more than one thread
  - Results stored in memory until instrumentation is ended

TSM Threads

- **TSM is multi-threaded !**
- **Server may have hundreds of threads active at a given time**
  - Use **show threads** for a list at any given time
- **Any given operation will likely make use of multiple threads**
- **Backup, for example, will use at least two threads:**
  - SessionThread – receives data from client
  - SsAuxThread – takes this data and passes to disk or tape
  - AgentThread – writes the data to tape
  - DiskServerThread – writes the data to disk
- **All threads can operate on different CPUs**



TSM Server Instrumentation

- **Started using server command:**
  INSTrumentation Begin [MAXThread=nnnnn]
- **Stopped using server command:**
  INSTrumentation End
- **Output generated when instrumentation is stopped**
- **Use the command line administrative client**
  dsmadmc -id=*id* -pass=*pass* inst begin
  dsmadmc -id=*id* -pass=*pass* inst end > *filename*
- **Use command redirection with storage agents**
  dsmadmc -id=*id* -pass=*pass agentname*: inst begin
  dsmadmc -id=*id* -pass=*pass agentname*: inst end > *filename*
- **Notes:**
  - Administrator must have system privilege

TSM Server Instrumentation Usage Strategy

- **Start server instrumentation just before starting the operation monitored**
  - For TSM 5.3, session and process numbers are included in the output if the thread is started while instrumentation is active
  - Using a TSM administrative client macro is an easy way to do this
- **Use for 1 to 30 minutes**
  - Careful! Long periods can generate a lot of information
  - Large number of threads makes it harder to diagnose a problem
- **Match up the multiple threads for a given session or process**
  - Use the session or process numbers in the instrumentation data (TSM 5.3)
  - Use the output of **show threads** command (during the operation)
  - Match the threads based on the amount of data moved
- **Look at threads with most of their time in areas other than 'Thread Wait'**
  - Most likely source of the problem

TSM Server Instrumentation Platform Differences

- **Instrumentation data is slightly different depending on platform**
- **z/OS**
  - Does not reuse thread IDs like other platforms
    - ✓ Thread IDs increase over time throughout server lifetime
    - ✓ Need to issue **show threads** command and note the current high water mark thread ID
    - ✓ Add 1000 to the high water mark, and use as the maxthread parameter on the inst start command
    - ✓ For example: **inst begin maxthread=5000**
- **UNIX**
  - Only 1 thread does I/O to any disk storage pool volume (called DiskServerThread)
    - ✓ Provides a disk volume centric view
    - ✓ May be harder to get complete operation disk statistics
- **Windows**
  - Any thread can do I/O on a disk storage pool volume (SsAuxThread for backup)
    - ✓ Provides a process/session oriented view
    - ✓ May be harder to see disk contention issues
  - Windows timing statistics only have about 15 millisecond granularity

TSM Server Instrumentation Categories

- **Disk Read** - reading from disk
- **Disk Write** - writing to disk
- **Disk Commit** - fsync or other system call to ensure writes are complete
- **Tape Read** - reading from tape
- **Tape Write** - writing to tape
- **Tape Locate** - locating to a tape block
- **Tape Commit** - tape synchronization to ensure data written from device buffers to media
- **Tape Data Copy** - copying data to tape buffers (in memory)
- **Tape Misc** - other tape operations (open, rewind, ...)
- **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** - something not tracked above

TSM Server Instrumentation Example

TSM thread id

Thread lifetime

```
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  Tottime  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
```

Overall Throughput

TSM Client Instrumentation

- **Identifies elapsed time spent performing certain activities**
- **Enabled using:**
  -testflag=instrument:detail (command line)
  testflag instrument:detail (options file)
- **Output is appended to a file in the directory specified in DSM_LOG environment variable**
  - For TSM 5.2, the **dsminstr.report** file
  - For TSM 5.3, the **dsminstr.report.p*PID*** file
- **Notes:**
  - Backup/archive client only (not in API or TDPs)
  - Command line client and scheduler only (not in GUI, web client)
  - Scheduler may need to be restarted after editing the options file
  - Cancel the client sessions from the server to get results without waiting for command completion

TSM Client Instrumentation Categories

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

TSM Client Instrumentation Example

TSM thread id

Thread lifetime

```
Thread: 118  Elapsed time   342.332 sec
Section      Actual (sec) Average(msec)  Frequency used

-----------------------------------------------------------------
Process Dirs         0.000          0.0               0
Solve Tree           0.000          0.0               0
Compute              0.673          0.0           47104
BeginTxn Verb        0.000          0.0              70
Transaction         59.315        847.4              70
File I/O           250.087          3.8           64968
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           19.004          0.4           47104
Confirm Verb         0.000          0.0               0
EndTxn Verb         12.443        177.8              70
Other                0.810          0.0               0

-----------------------------------------------------------------
```

# of transactions

Example 1

Problem Description:

- **z/OS server**
- **Customer just installed 3592 tape drives**
- **No other changes to their environment**
- **Unhappy with disk-to-tape storage pool migration performance**
  - 7-12 MB/sec per drive
- **Tape-to-tape storage pool backup performance is good**
  - 41 MB/sec per drive

Plan:

- **Rerun disk-to-tape storage pool migration test**
- **Collect server instrumentation data**
- **Collect show threads data**

Example 1 (cont.)

The "show threads" 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)
```

The instrumentation shows the performance of the migration thread reading the disk:

```
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
```

Large amount of time in Disk Read indicates the disk is the bottleneck.
Note: Divide the total KB read by the count to get the read IO size: ==> 256 KB.

Example 1 (cont.)

Problem Resolution:

- **Customer created new disk storage pool volumes:**
  - Using a newer disk subsystem
  - Using VSAM striped datasets
- **Now, disk-to-tape storage pool migration performance is good**
  - 37-40 MB/sec per drive

Example 2

Problem Description:

- **Windows server**
- **LTO 1 SCSI-attached tape drives**
- **AIX, Windows, Exchange clients**
- **Backup to tape is slow for all clients**
  - Backup of 30GB took 7 hrs (1.2 MB/sec)
- **Disk-to-tape storage pool migration performance is slow**

Plan:

- **Rerun disk-to-tape storage pool migration test**
- **Collect server instrumentation data**

Example 2 (cont.)

The instrumentation shows the performance of the migration threads:

```
Thread 61 DfMigrationThread (Win Thread ID 4436)  17:39:076-->17:47:38
Operation       Count  Tottime  Avgtime  Min-   Max-    Inst     Total
                                         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    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
```

Example 2 (cont.)

Problem Resolution:

- **Problem is clearly related to the tape subsystem**
- **Need to investigate the following:**
  - Tape attachment path
  - Tape drive device driver level
  - SCSI adapter driver level
  - SCSI adapter settings
- **Customer upgraded SCSI adapter device driver**
- **Now, disk-to-tape storage pool migration performance is good**
  - 19 MB/sec per drive
- **Client backups to tape are much faster**

Example 3

Problem Description:

- **AIX server**
- **Exchange client backup to tape is slow**
  - 240 KB/sec
- **LTO 1 and 3590 tape drives**

Plan:

- **Rerun backup test**
- **Collect server instrumentation data**
- **Note: This is an API client, so can't take client instrumentation data**

Example 3 (cont.)

The instrumentation shows the performance of the backup threads:

```
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 AgentThread 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
```

Example 3 (cont.)

Problem Resolution:

- **Problem is clearly related to receiving data from the client**
- **Could be a network or a client problem**
- **Customer measured FTP performance on the same route**
  - Shows a clear network problem
- **Reconfiguring the network improved throughput**

```
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>
```

Example 4

Problem Description:

- **AIX server**
- **Slow incremental backup of Windows clustered file server (39 hrs)**

```
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
```

Plan:

- **Rerun backup test**
- **Collect client instrumentation data**

Example 4 (cont.)

The instrumentation shows the performance of the backup threads:

```
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
-----------------------------------------------------------------
```

Large amount of time in EndTxn indicates a problem with the TSM server database. Noted the TSM database, recovery log, and disk storage pool volumes on same LUN.

Example 4 (cont.)

Problem Resolution:

- **Reconfigured the TSM server database to use multiple volumes on multiple LUNs**
- **Next incremental backup took 17 hours (2 times faster)**
- **Additional network configuration problems were found and fixed**
- **Customer implemented using journal-based backup**
- **Incremental backup now takes about 4 hours (10 times faster)**