Reporting TSM

Table of Contents

Reporting TSM

Introduction

The justification for this paper may be found in the great interest in the community of TSM users for the subject. This interest is the result of two forces, one, the relative complexity of the TSM backup/recovery application itself, and two, the fact that IBM does not currently provide with TSM a Reporting tool that is sufficiently comprehensive for large enterprise-level installations, *and* that is relatively simple to install and maintain. (IBM does offer a reporting system, which uses the methods discussed below and that generates reports that address much the same issues covered in this paper. However, this offering is itself expensive, complex and requires the installation and configuration of several components. IBM also now offers at no charge, a reporting component that is relatively easy to install and use, but does not offer the benefits of data capture for historical analysis and does not do the kind of in-depth analysis discussed in this paper. It is not surprising, therefore, that we find several commercial entries in the TSM Reporting market segment. The author is the primary architect of one such offering, and much of the discussion below is a result of several years of thinking about and building that tool.)

Reporting TSM

Why report?

There are at least three consumers of TSM reporting, and each has his or her requirements to consider. The TSM administrator has specific and general questions, answers to which are required to support the three common tasks an administrator of any system faces: strategic or planning tasks, tactical or maintenance tasks, and emergency or reactive tasks. The great majority of the reports created are aimed at the TSM administrator.

The TSM customer or client has much more focused requirements. Will the data he or she is responsible for be recoverable in the necessary time frame in the event the primary data becomes unavailable or unusable, and does the process of ensuring that recoverability interfere with the normal application environment beyond acceptable thresholds? As a consumer, the interesting metrics are Recovery-Time-Objectives and Recovery-Point-Objectives. A long RPO implies the need to recover data that is "old" in some sense, in hours, days, months or even years. A short RTO implies a special design in the storage pools that can support "rapid" recovery of the most current backup. A very short RTO of large amounts of data would normally be addressed with some sort of replication, for example, and not TSM.

Lastly, the managers responsible for the corporate data assets have their own, more abstracted level of requirements, generally focused on business continuance and regulatory issues. These have RTO and RPO targets as well, but they will be concerned with broader issues, such as the time to recover an entire data center at a DR site. In addition, managers may be interested in some sort of "chargeback" reporting- that is, the ability to apportion infrastructure costs to "clients" based on some kind of metrics that make sense from a TSM perspective. Additionally, managers will have interests focused on future budgetary requirements and, in some cases, in having the audit results to affirm corporate or governmental regulatory compliance.

TSM reporting in this paper will discuss how all these consumers can be provided the information they need to meet their individual requirements. It will be helpful to keep in mind the different roles these consumers play, as we begin to look at TSM reporting in detail.

Reporting TSM

What to report?

TSM is a complex application that can involve the entire data center structure, touch every hardware and software component, and is visible to every corporate level from data center Operations staff to the CEO, particularly when an abnormal event occurs. One result of this complexity is a wide range of possible reporting targets, and some sort of categorization will be necessary to allow a structured approach to the problem of what to report.

To that end, I will use a resource-centric view of TSM reporting, and I will divide the various resources TSM uses into four categories- TSM clients, networks, TSM hosts (the servers that host the TSM Server instances), and storage. Encompassing all of these categories is the dimension of time and its close relative, bandwidth. For example, two common reporting measurements are how long a particular client backup session ran, and how many bytes were sent. This information in isolation does not carry much value, so we need to know the surrounding circumstances, such as how much of the available window of time for backup on that client was used, was the data backed up sufficient to provide the level of recoverability required, were there any host storage problems during the session, and so on, to provide a truly valuable report.

Client reports will include backup and recovery, archive and retrieve, HSM migration and recall sessions, with the amount of data transferred, the elapsed time of the sessions, the network and tape resources used, and any abnormal events, such as missed or failed scheduled actions.

Network reports will include usage factors, such as bandwidth threshold saturations, communication errors, loss of contact, etc. IP networks as well as FC networks will need to be considered.

TSM server reports will include internal health checks, scheduled windows met or overrun, administrative processes successes and failures, and other related reports.

Storage reports will address the five kinds of storage critical to the TSM server application- the TSM database itself and its recovery log, disk storage pools, tape storage pools, and non-pool-based volumes (DB backups, Backupsets, Export tapes, and the like). Issues around tape libraries, such as slot occupancy, inter-site bandwidth consumption and offsite tape movement will also be important to consider.

While the bulk of this paper considers Reporting to be a daily procedure, there is a related kind of reporting which may be labeled Alerting. Alerting would provide near-time reporting of conditions that may require immediate attention by either a human operator or an automated process. Many of the conditions that are reported each morning may be candidates for Alerting in near-time. Which would be implemented is, naturally, a very site-specific kind of exercise. (In most cases, the time spent defining who would receive an alert and what they would do with it is better spent designing a more robust system in the first place.)

It also bears mentioning that, while most of the reporting discussed below focuses on each TSM server instance, some reporting can only be produced by considering the total environment. For example, it is not uncommon to have a single tape library being used by multiple TSM instances, either through TSM Library Sharing, or by library partitioning, or by some other method. It can be of great use to see a picture of that library's activities as a whole, as relative library load versus capacity is only visible from such a vantage point. Another example would be inter-site communication activity, either IP or FC based, seen as aggregate bandwidth used over the day, to point out saturation points being reached. Yet another area would be during migration of client data from one instance to another, typically using Export-Import functions. Finally, being able to see a summary of data across all TSM instances can be useful, particularly as a starting point for the day's administrative work.

Reporting TSM

What data to acquire?

In the normal TSM installation, there is but one basic repository of data pertinent to our reporting needs, and that is the TSM database itself. While there are other useful data sources, such as the TSM client log files, the TSM server accounting logs, O/S logs, external scheduler logs, ACSLS and EDT configurations and logs, and Z/OS SMF data, the great majority of useful information is to be found in the TSM database. (See Appendix C for a discussion of other sources.) This database is exposed to our view through a relational database model, via a Structured Query Language tool, the Select statement. The select statement can return data from tables, and TSM provides a rich collection of tables, describing object states and process events. (Appendix A lists many of the tables exposed to SQL selects, and with each new TSM release, new tables are generally added.) The problem of TSM reporting thus becomes one of deciding which tables to access, what the relationships of these tables imply, and how to present the data acquired for the three different types of consumers discussed above. (What kind of data can be used from the other sources will also be discussed below. Generally, some sort of scripting will be required to get and transform the source data into a form suitable for the TSM Reporting application to use.)

When deciding on which tables to query, at least two factors need to be considered- the affect on the TSM server and the level of aggregation desired for reporting. For instance, the volumeusage table contains data on how “fractured” a node is- that is, over how many tapes the data is spread. This table can be very large, however, and may require significant TSM server resources to extract. Another example is the backups table, which carries file-level information, but is almost always too large for daily queries to be feasible. When table size is a factor, it may be possible, by using the indexing information available, to issue several queries, each acquiring specific key-based data, to avoid some of the problems in one extended query. Aggregation can be achieved within the select statement itself using standard SQL clauses, or may be performed within the Reporting Tool.

Reporting TSM

How to acquire the data?

To issue a select statement against the TSM database, we will use one of the two available tools, the command line-based administrative client, named dsmadmc on windows systems (The other being access via the ODBC protocol.) To provide a flexible environment, we will assume this program will be run in a script, and the output will be in csv (comma-separated-values) format, suitable for parsing and direct use in tools such as spreadsheets. (Appendix B contains some sample scripts and outputs.) Since the data will need to be acquired periodically (once a day), we will assume the script is scheduled to run by an external task scheduler, and that the data is accumulated over time in files. Since the reports are meant to be viewed by humans, a daily cycle of data acquisition and reporting is assumed. To supplement this “snapshot” form of reporting, adhoc reports are usually produced on demand during the working day. (Alerts may be produced throughout the day, and the acquisition of data and the sending of alerts generally requires a site-specific customization.)

Since the Windows-based workstation is generally ubiquitous, it will be assumed the Reporting Host is such a platform. The discussion is not much altered, however, if the host is Unix or Linux based. (Shell scripts instead of command files, for example.) A good programming practice would be for the scripts to be platform-independent where possible, of course.

How do we transform data into reports?

Once we have acquired a day's worth of data from the various sources, it will need to be combined with the historical data from previous days and then transformed into the reports that our consumers will need. While an in depth discussion of this analysis is beyond the scope of this paper, an outline may prove of interest to the reader, and is the topic of this section. We will refer to the reporting application as the Reporter and the TSM server being reported on as the Server.

The first step is establish the initial conditions of the Server, as saved in the previous day's reporting run as final conditions. This would include what processes were running, what tapes were mounted in which drives, what sessions were running, and any other significant items. For sanity checking, the Reporter should check that the saved final conditions do in fact correspond to the initial conditions as acquired from the Server. Many causes could result in a mismatch. For example, if the Server was halted and restarted, most activities would not survive!

The second step would be to process historical data of interest, to provide a baseline of averages and exceptions that will form a context for much of the reporting to come. For example, reporting that the DB backup took 60 minutes does not tell us much, but if we add the information that the average time over the last 30 days was 30 minutes, it may now be of interest. Which data and how far back to look is a function of the comprehensiveness of the Reporter. With the rate of change of technology, of the functionality of TSM, and of the customer's environment, the usefulness of most trend data is relatively limited now compared to some years back. For instance, since most tape drive technology is being replaced on a schedule of a few years, tape capacity and throughput statistics lose their usefulness on that scale. Client workloads may change drastically on a scale of months, and thus that data may have a shorter useful life.

An important part of the Reporter's job is to create new "meta" data, and save it for analysis. This is to provide efficiencies in some cases, and to create new information resources in other cases. An example of the former would be a file with node occupancies by date. The TSM table occupancy is by nodename-filespacename, and it is sometimes more convenient and efficient to aggregate by just nodename. (You can use aggregation phrases in the select statement sent to TSM, but that puts an extra load on the TSM server, and you lose the capability of using the detail data for other purposes.) Two examples of the latter would be a file with I/O errors by library/drive and datetime, and a file with tape throughputs by process-type, and datetime.

The third step would be to process all the "static" table-based information we just acquired. This includes things like storage pools, nodes, libraries, drives, options, etc. Some of these will be used to determine deltas, or changes from yesterday. A good example would be storage pool occupancy, which reflects the net activity of backups, archives, migrations and expirations (and move data commands, and delete filespaces, and ….) Others will serve as look-ups. An example of this would be the devclasses table to cross-reference storage pools to libraries. A reasonable count of the number of different tables acquired from a TSM instance would be on the order of four dozen.

With the initial conditions, the historical precedents, and the day-ending status of the Server in hand, we are now ready to examine the day's activities. The primary source of the day's activity is the Activity Log, or ACTLOG. This is supplemented by the summary table or possibly by the Accounting Log, since some TSM clients do not report sufficient information to be posted to the ACTLOG. A large part of the complexity in reporting a Server is connected with the vast differences in both the syntax and content of various TSM client-generated ACTLOG messages (basic, API, DP, etc.) and the changing syntax and content of TSM Server-based ACTLOG messages (by version-release-level-patch.) Just parsing the imbedded quotes and continuation lines is a challenge when dealing with the ACTLOG!

As each ACTLOG message is processed, several things happen. The message in itself is examined and categorized. (The number of categories is important. Too many categories and the categorization process quickly loses its purpose. Too few, and the result is meaningless. TSM itself tries to categorize by informational, warning, and error, which is too few in my opinion. I prefer about seven: Catastrophic, Client B/A, Library, Admin Processes, Server Maintenance, Informative, and System Integrity.) If the ACTLOG message raises a condition on its own, that condition is raised. If the message relates to an existing process or session, any pertinent information in the message is saved for the related event. If the message establishes a new process or session, that event is initialized. If the message terminates a process or session, that event is flagged, and any runtime statistics are accumulated. Some messages can do all three- a Server restart, for example, initiates a new Server "run", terminates all sessions and processes that were active, and raises the condition "Server Halted/Started".

Some messages require a level of "guessing" to relate to the appropriate event. Older TSM versions, for example, did not label tape mount messages with the connected process or session. A guess would need to be made as to which active process or session most likely caused the tape mount. Sometimes this guess will be corrected later on, say, when it is discovered a tape volume was the output of a DB Backup.

Once all the ACTLOG messages have been processed, report generation may begin. The first step is to check for conditions to be raised by things that didn't happen. For example, if no DB Backup was detected, that might be a condition of interest. Also, if an externally scheduled important client backup was not detected, that too may be of interest to someone. Next, certain "best practice" kinds of configuration are checked. An example would be to compare the reuse days for copy storage pools to the retention days for DB Backups. Or to check for administrative ids whose passwords are due to expire. Or to check for combined times for Expiration and DB Backup exceeding two hours. There is a seemingly endless supply of these kinds of conditions. While a small number may be identified by an experienced TSM admin off-the-cuff, this list is best maintained by watching many different installations over time, and talking with as many TSM administrators as possible.

Reporting TSM

Broadly speaking, the reports generated each day fall into six groups, each with its own sub groupings. A statically-oriented Reporter will generate all of the reports for each Server every day, while a dynamically-oriented Reporter will wait and generate the report on-the-fly when asked. (A dynamic Reporter may also prepare some reports by schedule, and a static Reporter may be able to produce some reports in a more ad hoc manner.) The advantages and disadvantages may be argued at length (and are!), but such arguments resemble those of the GUI vs command line debates, in that they approach non-rationality rapidly.

Reporting TSM

Reports for the TSM Server Administrator

Conditions – (Categorized and weighted)

- Bad Things / Noteworthy Events
  - Configuration
  - Administrative
  - Library
  - Client
  - Other
- Good Things (To cheer you up)

Admin Processes

- Schedules
- Customer-defined Windows (with compliance results)
- Processes That Ran (with results)
- Commands Issued (with results)

Client Sessions

- Detailed by session
- Summaries by various keys (Domain, platform, activity, node, etc.)
- Top ten lists (Faster/slowest/biggest/oldest/etc)
- Upgraded/downgraded clients
- Nodes without (you name it, schedules, mountpoints, etc) or with (tape mounts, API activity, etc.)
- Throughputs by time-of-day

Library-related

- Tape activity by time-of-day
- Drive occupancy by time-of-day
- Tape volume status changes
- Tape drive throughput
- Tape volume utilization
- Tape consumption
- Client use of tapes
- Global library reports
  - Drive occupancy of shared libraries
  - Drive Error History
  - Link Bandwidth consumption
    - By site (location)
    - By Virtual Library

Reporting TSM

Workload-related

- Storage pool sizes and workload
- Throughput by time of day
- Unusual occupancy changes
- Migrations
- Copy Pools
- Reclamations
- Move Data
- Capacity Sizing (of all the main resources)
  - Based on today's numbers
  - Based on recent historical trends

Charts

- Anything you can think of
  - Today by time-of-day
  - By-day 90 days back
  - By day-of-week
- Charts per TSM instance and "global" charts across one or more instances

Reporting TSM

Reports for the TSM Client Administrator

Detail Client Sessions
Object Errors
Missed / Failed Schedules
Session History

Reports for Corporate Storage Managers

Daily Summaries – Workload and Resources Used
Weekly Summaries – Workload and Resources Used
Monthly Summaries – Workload and Resources Used
Charge Backs
Capacity Sizing
Charts

Reporting TSM

Things IBM could do to make life easier

With each release of TSM, IBM has made life better for the Reporting Tool author. A good example is the message now found in the ACTLOG showing who mounts a tape-ANR0510I. This message significantly reduces the "guesswork" needed to assign a mounted tape to an activity. Another recent addition is showing the session number and process number that are related to a message.

However, there are a few things IBM could still do that would simplify our work lives. What follows is a short list of what I would want most (in no particular order). (Some of these may actually already be in versions I have yet to encounter.)

- Show mount messages for Z/OS-based TSM servers
- Make it an option to expose a table indexed by node and management class and containing object-level attributes (size, active/inactive, date)
- Publish the Backup-Archive Client API
- Allow a BA Client to run read-only as a virtual node running on a "virtual" platform: (dsmc –virtualnode=abc –virtualplatform=Netware)
- When running expiration, display the occupancy reduction by node/filespace
- In the dismount message, show elapsed time, time-to-mount, transfer-time
- Enforce some sort of minimal posting to the ACTLOG by API clients
- Provide a client option with text to post to the ACTLOG upon session open/close (sort of a poor-man's ISSUE MESSAGE feature)
- Provide some user-defined columns to the NODE table

•

Appendix A TSM Tables

ACTLOG	Server activity log
ADMINS	Server administrators
ADMIN_SCHEDULES	Administrative command schedules
ARCHIVES	Client archive files
AR_COPYGROUPS	Management class archive copy groups
ASSOCIATIONS	Client schedule associations
AUDITOCC	Server audit occupancy results
BACKUPS	Client backup files
BACKUPSETS	Backup Set
BU_COPYGROUPS	Management class backup copy groups
CLIENTOPTS	Client Options
CLIENT_SCHEDULES	Client schedules
CLOPTSETS	Client Option Sets
CONTENTS	Storage pool volume contents
DATAMOVERS	Data Movers
DB	Server database information
DBBACKUPTRIGGER	Database backup trigger information
DBSPACETRIGGER	Database space trigger information
DBVOLUMES	Database volumes
DEVCLASSES	Device Classes
DISKS	Disks
DOMAINS	Policy domains
DRIVES	Drives
DRMCSTGPOOLS	Copy storage pools managed by DRM
DRMEDIA	Physical volumes managed by move drmedia
DRMMACHINE	Disaster recovery manager machine information
DRMMACHINECHARS	DRM machine characteristics
DRMMACHINENODE	DRM machine node associations
DRMMACHINERECINST	DRM machine recovery instructions
DRMMACHINERECMEDIA	DRM machine recovery media associations
DRMPSTGPOOLS	Primary storage pools managed by DRM
DRMRECOVERYMEDIA	DRM recovery media
DRMSRPF	Recovery plan files in source server
DRMSTANZA	Recovery plan file stanza names
DRMSTATUS	DRM status information
DRMTRPF	Recovery plan files in target server
EVENTS	Scheduled Event Results
FILESPACES	Client file spaces
GROUP_MEMBER	Group Members
LIBRARIES	Libraries
LIBVOLUMES	Library volumes
LICENSES	Server feature licenses
LICENSE_DETAILS	License usage details

LOG	Server recovery log information
LOGSPACETRIGGER	Recovery Log space trigger information
LOGVOLUMES	Recovery log volumes
MEDIA	Physical volumes managed by move media
MGMTCLASSES	Management classes
NODES	Client nodes
OCCUPANCY	Client storage occupancy
OPTIONS	Server Options
PATHS	Paths
POLICYSETS	Policy sets
PROCESSES	Client schedule associations
PROFILES	Profiles
PROF_ASSOCIATIONS	Profile associations
RECLAIM_ANALYSIS	Server reclamation analysis table
RESTORES	Client restore operations
SAN	SAN attached devices
SCRIPTS	Server Command Scripts
SCRIPT_NAMES	Server Command Script Names
SERVERS	Remote server nodes
SERVER_GROUP	Server Groups
SESSIONS	Active client sessions
SPACEMGFILES	Client space-managed files
STATUS	Server status
STGPOOLS	Storage pools
SUBSCRIPTIONS	Subscriptions
SUMMARY	Server Activity Summary Table
VOLHISTORY	Volume history information
VOLUMES	Storage pool volumes
VOLUMEUSAGE	SEQUENTIAL volume usage by client node

COLUMNS	SQL Table Column Catalog
ENUMTYPES	SQL Enumerated Types Catalog
TABLES	SQL Table Catalog

Appendix B Sample scripts to acquire data from TSM tables

A script to generate the list of tables in Appendix A-

```
dsmadmc -id=admin -pa=admin -comma SELECT TABNAME, REMARKS from
        SYSCAT.TABLES
```

A select statement to get a list of columns defined for a particular table. Note that each column may a part of an index to the table. See the entries INDEX_KEYSEQ and INDEX_ORDER.

```
SELECT * FROM SYSCAT.COLUMNS where TABNAME='SUMMARY'

        TABSCHEMA: ADSM
        TABNAME: SUMMARY
        COLNAME: START_TIME
        COLNO: 1
        INDEX_KEYSEQ: 1
        INDEX_ORDER: A
        TYPENAME: TIMESTAMP
        LENGTH: 0
        SCALE: 0
        NULLS: FALSE
        REMARKS: Start time
```

A select statement to select certain columns from a table-

```
SELECT START_TIME, END_TIME, ACTIVITY, ENTITY, AFFECTED, BYTES
        from  SUMMARY where date(START_TIME) = '08/14/2005'

        START_TIME: 2005-08-14 00:00:08.000000
        END_TIME: 2005-08-14 00:00:25.000000
        ACTIVITY: BACKUP
        ENTITY: NJTSMAIX
        AFFECTED: 53
        BYTES: 15557850

        START_TIME: 2005-08-14 23:00:13.000000
        END_TIME: 2005-08-14 23:02:23.000000
        ACTIVITY: EXPIRATION
        ENTITY:
        AFFECTED: 624
        BYTES: 0
```

Appendix C Other sources of data

TSM server accounting log

In some older versions of TSM, data from the accounting log was more dependable than data from either the Summary table or the Actlog. This file is a csv-formatted file, usually maintained on the TSM instance host disk, and would need to be transferred to the Reporting Host, or otherwise made accessible. The layout of the Accounting log is defined in the Administrator's Guide (Chapter 20, in **Monitoring IBM Tivoli Storage Manager Accounting Records**, in the v5.3 book.)

TSM Client dsmched.log, dsmerror.log, dsierror.log, etc.

The TSM client logs provide detail that is sometimes not available to the TSM server instance. The challenge to using this data is getting the logs (or a processed version of them) to the Reporting Host at the right time. Then a script can parse the files and select events and statistics of interest for use by the Reporter.

Z/OS SMF data

At least two kinds of data important to Reporting are kept in SMF files on a Z/OS-based TSM server. Current and past TSM releases do not reflect tape mounts in the Activity log, and so this information must be gleaned from the SMF record types 14 and 15. (Dismounts are reflected in the Actlog.) Data that normally is kept in the Accounting Log file is kept in SMF records of type 42. To make the SMF data accessible to the Reporting Host, the data must be filtered from other SMF records and then converted from EBCDIC to ASCII, and finally made available to the Reporter.

ACSLS configuration and logs

ACSLS configuration information is normally acquired by running command line scripts on the ACSLS host system. The output files are then transformed into a usable format and transferred to the Reporting Host. The important data includes library volume detail and scratch pool information. ACSLS logs may also be made available to the Reporting Host for alerting as well as reporting uses.

EDT configuration and logs

When Gresham EDT is used for library sharing in an ACSLS-supported library environment, some of the data normally abstracted from the ACSLS configuration is instead acquired from the EDT configuration files. Other data extracted includes the information normally found in the TSM paths table (device-to-drive cross-reference.)

O/S logs and performance tracking files

Sometimes it is convenient to be able to report TSM performance statistics in conjunction with the TSM server host system statistics such as CPU utilization or disk activity. A simple scheduled script can query the O/S periodically and save the data in an easy-to-use format, such as CSV (comma-separated-values). These files can then be use by the Reporting application.

Manually Entered Information

Some configuration information that is vital for TSM reporting is just not available by automated means, and must be manually acquired and maintained. This includes such items as Server and Library geographic locations, IP Network masks and adapter allocation, HBA zoning, and all of the customized reporting, charting, and alerting settings that are desired by the TSM reporting consumers. Someday, open protocols may allow some of the SAN/LAN data to be acquired through SRM tools, and thus reduce the manual maintenance load. And, for those GUI-enamored folks, front-end programs may make the data entry process more palatable.

Reporter-generated Metadata

Data abstracted and regrouped by the Reporter each day can be saved and used in the future to support trend computations. It is a vital link in the cohesiveness of the sequential reports each day.