



IBM Software Group

TSM HSM Explained



Oxford University TSM Symposium 2003

 business on demand software

Christian Bolik (bolik@de.ibm.com)
IBM Tivoli Storage SW Development

1

© 2003 IBM Corporation

IBM Software Group | Tivoli software



Agenda

- Introduction
- How it works
- Best Practices
- Futures
- References

2

TSM HSM Explained | Oxford Symposium 2003 |

© 2003 IBM Corporation

Introduction



TSM HSM and e-business on demand

Goals of @-business on demand:

- Computing as a utility
- Integrate business processes
- Increase utilization of existing assets



TSM HSM provides:

- File-level storage virtualization
- Active data is retained on online storage
- Inactivate data is moved to near-line storage, Cost Savings making space available on online storage
- Faster, "on demand" restores

Definition of HSM (the TSM-way)

- Current official name is “IBM Tivoli Storage Manager for Space Management” (others were “Tivoli Space Manager”, “ADSM HSM”, ...)
- HSM automatically migrates files from on-line to near-line storage (typically from disk to tape), based on policy settings
- Small “stub files” are retained on disk, appearing as the original files, thus ensuring transparency of HSM for user applications
- Stub files contain information pointing to corresponding entry in TSM server DB (using an opaque “id”, thus independent from stub’s path)
- Migrated files are recalled automatically back to disk as required

Note: TSM server has its own type of HSM for migration of files in its storage pool hierarchy

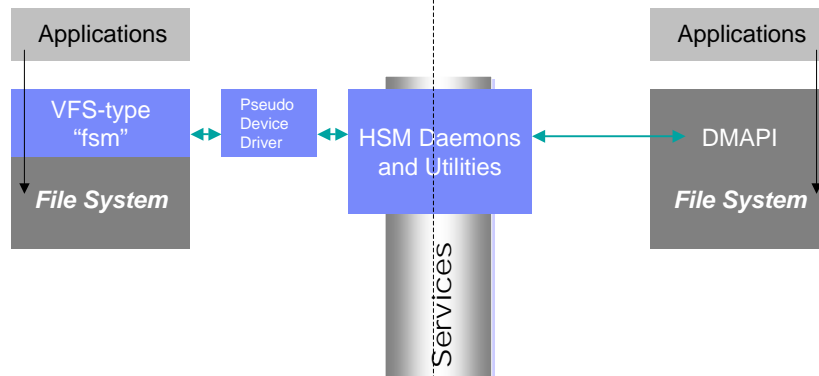
TSM HSM Reconciliation

- Main purpose since TSM HSM V4.1.2: Release TSM server storage for migration copies of deleted/modified files
 - ▶ Determination of migration candidates now done by separate “dsmscoutd” daemon
- Is one of the more “controversial” features of TSM HSM:
 - ▶ Makes TSM HSM more space-efficient than any other UNIX HSM:
 - obsolete file copies are removed from server storage
 - other HSMs may use these copies as backup substitutes
 - ▶ Used to be a very expensive operation since full tree traversals were required to compare local file system and server namespaces
- Since 4.1.2 full tree traversals in most cases no longer required, HSM maps directly from server namespace to local filesystem
 - ▶ Exception: full tree traversal for now still required once after stubs restored using “dsmc restore –restoremigstate=yes”

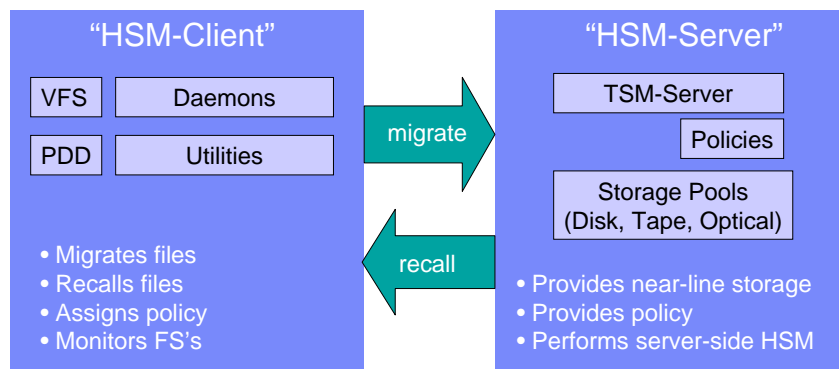
TSM HSM comes in 2 flavors:

HSM for IBM AIX JFS:
VFS-based

HSM for IBM AIX GPFS and
Sun Solaris VxFS: **DMAPI-based**



HSM as a Client/Server application





IBM Software Group

How it works

Tivoli software

business on demand software

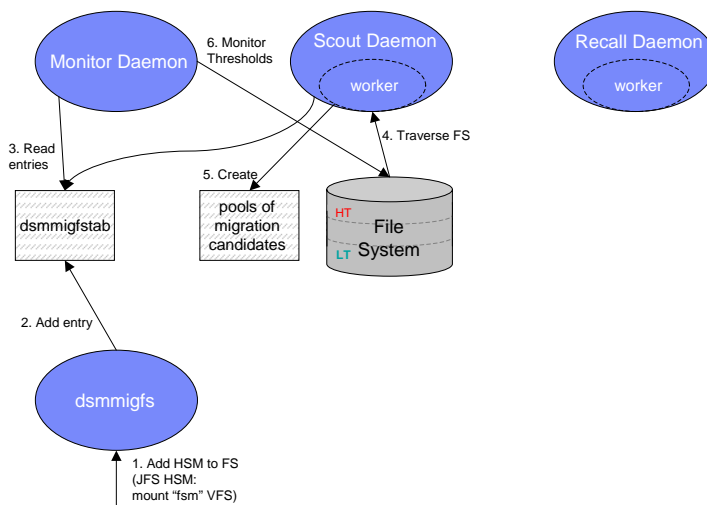
9

© 2003 IBM Corporation

IBM Software Group | Tivoli software



Act 1: Adding HSM

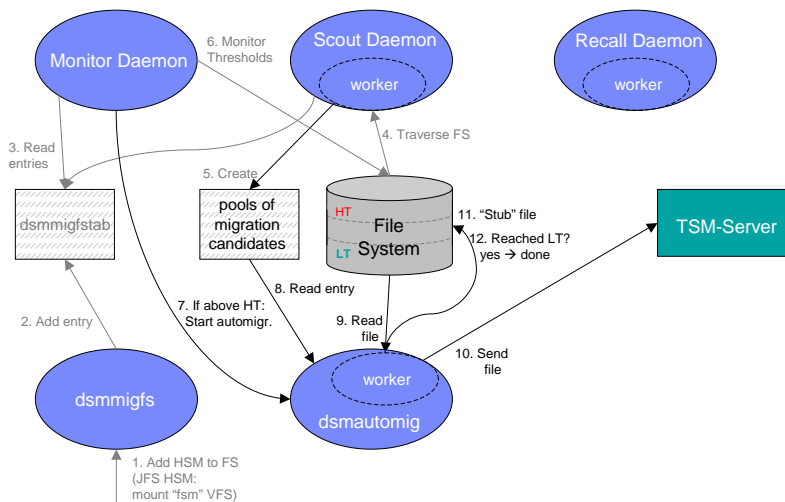


10

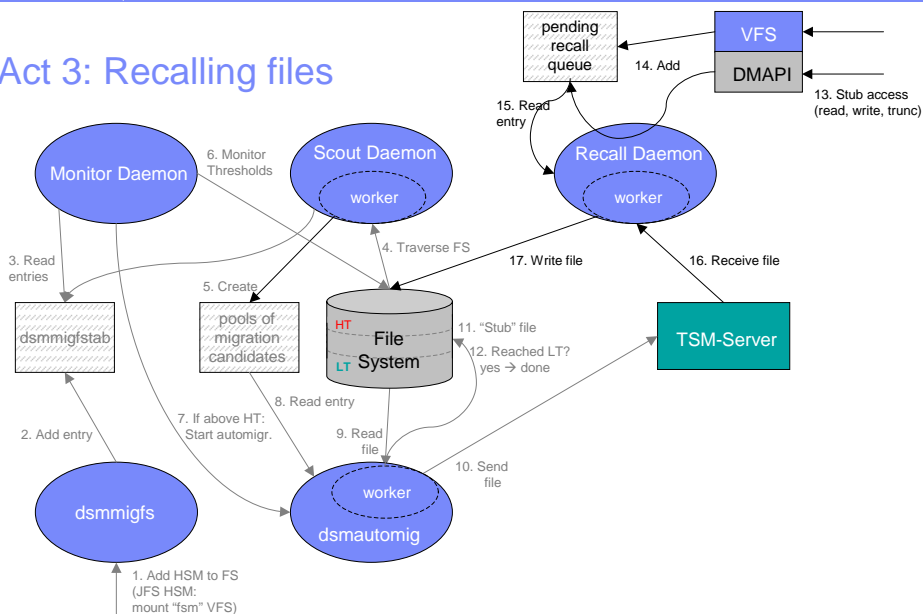
TSM HSM Explained | Oxford Symposium 2003

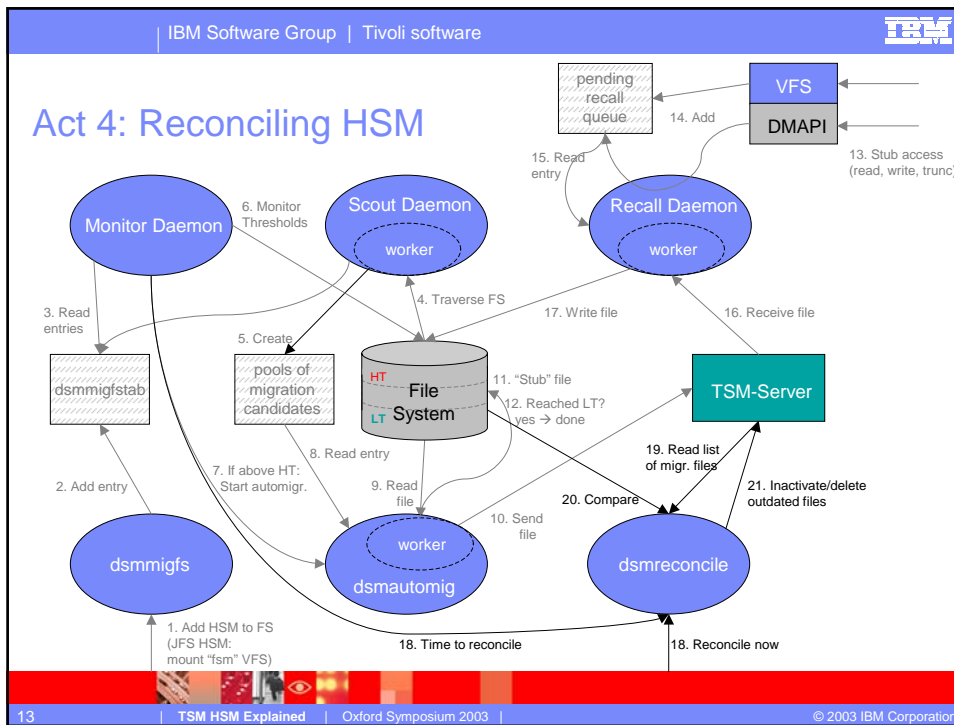
© 2003 IBM Corporation

Act 2: Automigration



Act 3: Recalling files





IBM Software Group

TSM HSM's use of the DMAPI

Tivoli software

business on demand software

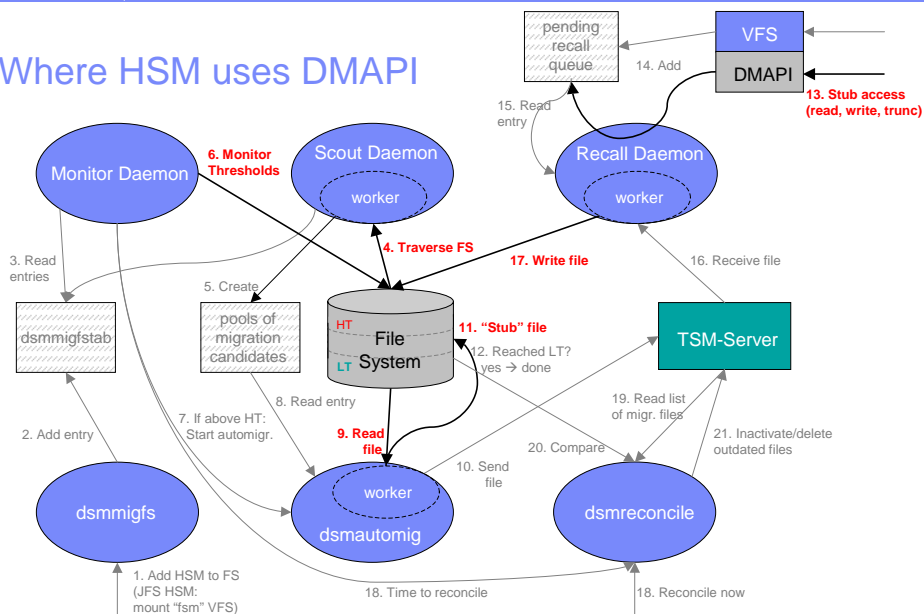
14

© 2003 IBM Corporation

DMAPI: What it is

- DMAPI is an open standard defined 1997 by The Open Group
- Official name: "Data Storage Management (XDSM) API"
- Goal: Independence of data management applications from underlying file system types
- Avoidance of kernel code by applications using DMAPI
- DMAPI in general provided by file system implementation
- Examples of file systems supporting DMAPI:
 - ▶ XFS for SGI IRIX and Linux, GPFS for IBM AIX and Linux, Veritas VxFS for Sun Solaris, JFS for HP-UX, JFS2 for AIX 52B

Where HSM uses DMAPI





DMAPI-based vs. VFS-based HSM

- Pros:
 - ▶ all user-level code (almost), lower maintenance cost
 - ▶ automatic generation of file-/file-system-related events
 - ▶ large portion of code can be reused across DMAPI implementations
 - ▶ DMAPI has broad adoption by file system and storage management vendors

- Cons:
 - ▶ files need to be staged to disk before user app can access them
 - ▶ standard contains mandatory and optional parts
→ some implementation differences



TSM HSM Best Practices



System Factors on HSM

- Number of managed file systems:
 - ▶ The more, the higher the HSM work load (monitoring, etc.)
- Mix of file sizes:
 - ▶ The more large files, the faster HSM finds "good" migration candidates
- Directory structure:
 - ▶ "Flat" structures in general are traversed more quickly
- Number of files in a given file system:
 - ▶ Affects time required for a "full" reconcile, e.g., after stubs were restored
- Rate of file creation/recalls:
 - ▶ The higher the creation rate, the more often automigration needs to run
 - ▶ The higher the recall rate, the higher the probability of getting into a "thrashing" situation
 - May be alleviated using management class parameter "automignonuse" (minimum days since last access), but need to have sufficient "old" files

Tips and Best Practices



- Place primary HSM storage pool on disk, with a "next"/secondary storage pool on tape (or optical) to avoid tape drive contention
 - ▶ Exploits TSM's server-side HSM
 - ▶ Set "cache" option of disk stg pool to "yes"
- Adjust HSM system-wide and file system-specific options according to environment, e.g.:
 - ▶ Value of MAXCANDPROCS should match number of managed file systems for maximum concurrency
 - ▶ Many concurrent requests to migrated files: value of MAXRECALLDAEMONS should be increased
 - ▶ see the "HSM Field Guide" for more on this
- Create/read-only file systems don't require reconciliation between file system and TSM server (set RECONCILEINTERVAL to 0)

TSM HSM and Backup



- HSM is not a backup solution:
 - ▶ When HSM migrates a file, the file is essentially “moved” rather than “copied”
 - ▶ Need to take care to always have at least 2 copies of each file
- HSM is integrated with TSM Backup and Restore:
 - ▶ “Inline backup” when backing up migrated files to same TSM server, migrated files are not recalled during backup (unless file has ACL set)
 - ▶ Files can be prevented from being migrated if no current backup copy exists (using policy setting “migrequiresbkup”)
 - ▶ Migrated and premigrated files are by default restored to stubbed state, helps cut down on restore time when restoring entire file systems
- Backing up a modified file will update its “last access” timestamp
 - ▶ Option PRESERVELASTACCESSDATE introduced in TSM 5.1.5

“On Demand” Restore

- HSM can be used to accelerate restores significantly
- Files that were migrated or premigrated are restored to “empty” stubs (without any file data => no tape mounts necessary)
 - ▶ Requires “-restoremigstate=yes” on “dsmc restore” command (default)
- Even faster: dsmmigundelete
- Simply recreates stubs for all the (pre-)migrated files the server knows about
- Drawbacks with respect to dsmc restore:
 - ▶ Resident files/empty directories not restored
 - ▶ ACLs not restored (dsmc restores complete file if ACL present)
 - ▶ “Point in time” restores etc. not possible
 - ▶ File renames/moves after migration may not be captured
- See HSM Field Guide for more details

TSM HSM Futures



Some "Past Futures" since 2000

- 3.7.2 (GA'ed 04/00)
 - ▶ Introduced file system-specific "maxcandidates" parameter to limit candidates search (trade-off speed vs. "best" candidates)
 - ▶ Allowed overlapping of dsmreconcile and dsmautomic runs
- 4.1.2 (GA'ed 12/00):
 - ▶ Concurrent auto-migration of multiple files per file system
 - ▶ Dedicated subprocess for candidates determination (dmscout), scans only until set number of candidates found, resumes where left off
 - ▶ dsmreconcile "immediate synchronization" avoids full tree traversal in most cases (except after restore with "restoremigstate = yes")
 - ▶ Full concurrency of dsmreconcile and dsmautomic

“Past Futures” (2)

- 4.2.1 (GA'ed 3Q01)
 - ▶ Candidates determination now controlled by separate daemon (dsmscoutd)
 - ▶ Added MAXCANDPROCS, CANDIDATESINTERVAL options to better control candidates determination process
 - ▶ New MINMIGFILESIZE option to specify minimum size of migratable files
- 5.1.0 (GA'ed 1Q02)
 - ▶ Support for fail-over of HSM for AIX JFS for HACMP
- 5.1.5 (GA'ed 3Q02)
 - ▶ User exits for HSM for AIX JFS (stubnotify, recallstart, recallend)
- 5.2.0 (GA'ed 1Q03)
 - ▶ Streaming recall mode for HSM for AIX JFS and Sun Solaris VxFS
 - ▶ Non-root users can run HSM utilities, same as for AIX JFS HSM

TSM HSM Roadmap 2003/2004

- Scheduled for 4Q03:
 - ▶ Basic HSM support for HP-UX JFS (DMAPI-based)
 - ▶ Basic HSM support plus streaming recall mode for Linux/x86 GPFS (DMAPI-based)
 - ▶ LAN-free HSM migrate/recall for AIX GPFS
 - Requires TSM Storage Agent to be installed on HSM nodes
 - Supported for sequential media only (i.e. not for disk stg pools)
 - ▶ Partial file recall for AIX GPFS
 - Equivalent to TSM API's "partial object restore" function
 - Recalls only requested portion, plus some "read-ahead" buffer to exploit tape streaming mode
 - Can delay need for threshold/demand migration

Dates and content subject to change based on business decisions

TSM HSM Roadmap 2003/2004 (2)

- Scheduled for 2H04:
 - ▶ Distributed (“multi-node”) HSM for AIX GPFS:
 - Automatically distribute migrates/recalls across multiple nodes in GPFS nodeset
 - Remote “mover” nodes centrally controlled by dsmautomic and dsmrecalld
 - ▶ LAN-free HSM migrate/recall for Linux GPFS
 - ▶ Partial file recall for Linux GPFS
 - ▶ Distributed HSM for Linux GPFS
 - ▶ New Java-based HSM GUI

Dates and content subject to change based on business decisions

Potential Future Enhancements

- Platform support:
 - ▶ TSM HSM for Windows NTFS, AIX 5.2 JFS2, Linux/JFS (xSeries, zSeries)
 - ▶ Solaris cluster, Veritas cluster
- Product integration:
 - ▶ “Active SRM” (integrate HSM and SRM)
 - ▶ Extend TSM server’s “simultaneous write” capability for immediate backup-on-migrate/premigrate-on-backup
 - ▶ Inline migrate: clone from backup (opposite of inline backup)
 - ▶ Premigrate/look for migration candidates during backup
 - ▶ Enable “on demand” restore for resident files also (by “recalling” their data from backup storage pool)
 - ▶ Auto-backup if “migrequiresbackup” set, rather than skip

Dates and content subject to change based on business decisions

More Potential Future Enhancements

- Capability:
 - ▶ Auto-inactivate (pre-)migrated copies on write
 - ▶ More flexible/customizable migration candidate criteria
 - ▶ Migrate-on-close recall mode for DMAPI-based HSMs
 - ▶ Reporting: better audit trail capabilities, capture number of automigrations over time (history), file-level "thrashing" report
 - ▶ "Striping" of very large files across multiple tapes (parallel transfer of multiple file segments)

...what else?

Dates and content subject to change based on business decisions

References

Tivoli software

business on demand software



References

- TSM HSM on the Web:
 - ▶ <http://www-3.ibm.com/software/tivoli/products/storage-mgr-space/>
- Tivoli Field Guide on HSM (from January 2003):
 - ▶ <http://www-1.ibm.com/support/entdocview.wss?rs=0&uid=swg27002498>
- DMAPI specification:
 - ▶ <http://www.opengroup.org/onlinepubs/9657099/>
- e-business on demand:
 - ▶ <http://www.ibm.com/ondemand>
- SNIA shared storage model:
 - ▶ http://www.snia.org/tech_activities/shared_storage_model



TSM HSM Explained



Oxford University TSM Symposium 2003



Christian Bolik (bolik@de.ibm.com)
IBM Tivoli Storage SW Development



IBM Software Group

Backup Slides

Tivoli software

business on demand software

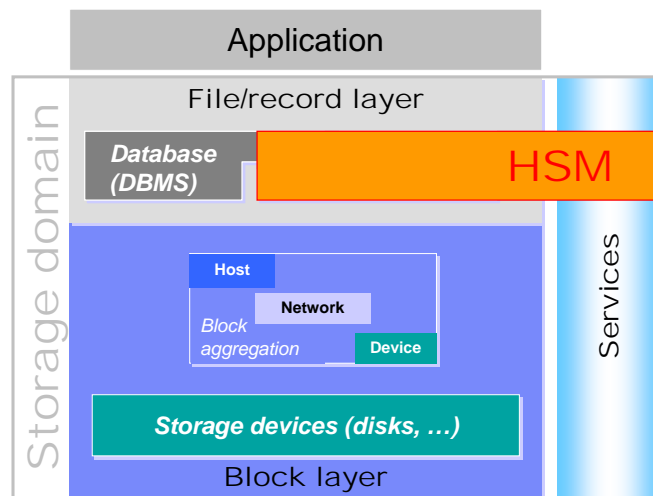
33

© 2003 IBM Corporation

IBM Software Group | Tivoli software



TSM HSM and the SNIA Shared Storage Model



34

TSM HSM Explained | Oxford Symposium 2003

© 2003 IBM Corporation

DMAPI: Concepts and terms

- **“Sessions”:**
 - ▶ DMAPI is session-based, to manage resources such as event queues
- **“Events”:**
 - ▶ DMAPI applications can subscribe to be notified when specific events in the file system occur (→ *HSM recall*)
- **“Holes”:**
 - ▶ DMAPI applications can request for a portion of a file to be released (→ *HSM stub file creation*)
- **“Managed regions”:**
 - ▶ Portions of files for which events should be generated upon access, as specified by the DMAPI application (→ *access to HSM stubs*)
- **“Handles”:**
 - ▶ In DMAPI all files are referenced by unique handles, not by path/name

How HSM uses the DMAPI

4. Traverse FS	Scout Daemon does bulk processing of dirs	dm_get_dirattr, dm_get_dmattr
6. Monitor Thresholds	Monitor Daemon registers for DMAPI “NOSPACE” event	dm_set_disp, dm_get_events
9. Read File	Dsmautomig reads using DMAPI function, avoids time stamp updates	dm_read_invis
11. Create Stub File	Dsmautomig “punches hole” into file, sets managed region	dm_set_region, dm_punch_hole, dm_set_dmattr
13. Stub Access	Recall Daemon receives event if managed region/stub is accessed	dm_set_disp, dm_get_events
17. Write File	Recall Daemon bypasses DMAPI event mechanism	dm_write_invis